

1982. MÁJ. 1. 9.

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest



MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

РГ-II / КНБВТ
"СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ И ИНФОРМАЦИОННЫЕ
СИСТЕМЫ"

СБОРНИК
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИХ РАБОТ

ТОМ I

A kiadásért felelős:

DR VÁMOS TIBOR

ISBN 963 311 128 5

ISSN 0324 2951

Főprint nyomda 82029

С О Д Е Р Ж А Н И Е
Сборник научно-исследовательских работ
рабочей группы РГ-11, КНВВТ

выпуск 131 - том I

Предисловие	7
Александров, А.П.	
Проблемы создания вычислительных центров коллективного пользования в Комитете по Единой системе социальной информации	11
Барнев, П., Кр.Марков	
Проблемы управления в системах информацион- ного обслуживания коллективов	15
Батурина, Л.Н., Н.А.Лепешинский	
Имитация работы сетей ЭВМ	25
Бельке, В., Х.-Д. Хартманн, Б.Лойхт	
Автоматизированная система классификации систем управления базами данных (СУБД)	31
Боянов, К.Л., В.С.Гетов, Х.А.Турлаков	
Высокопроизводительные параллельные процес- соры с сетевым программированием	39
Денев, И.Д., Е.К.Живкова, Р.П.Лесева	
Средства форматирования	53
Добрев, Д.М., Р.К.Киркова, П.А.Парванов	
Система планирования и учета вычислительных ресурсов	57
Добрев, Д.М., Й.В.Швертнер	
Доступ к записям в СУБД БИСЕС	63
Жечковски, В.	
Язык манипулирования данными системы управ- ления базой данных LINDA	67
Занев, В.	
Концепции и модель распределенной системы информационного обслуживания коллективов	79

Златарова, Ф.	Эквивалентность моделей данных в распределенных базах данных	89
Игиркова, Р. К.	Применение информационных систем при организации научных мероприятий	99
Игиркова, Р. К.	СОКРАТ - Система для оперативного контроля над реализацией и отчетом заграничных служебных командировок	105
Кондратьев, А. И.	Подход к построению математической теории для классов информационных систем	111
	Список мероприятий РГ-11 "Системы управления базами данных и информационные системы"	123
	Имена и адреса участников РГ-11	125

выпуск 132 - том II

Кузнецов, Г. И.	Некоторые вопросы обработки информации в условиях ВЦКП СО АН СССР	7
Ласкин, М. Ф., В. И. Безряков	Об использовании мини-ЭВМ в системе ВЦКП	21
Либл, Л.	Базы данных и их назначение в процессе сбора и обработки информации	31
Марчук, Г. И., О. В. Москалев	Проблемы и эволюция вычислительных центров коллективного пользования	39
Метляев, Ю. В.	Технические средства ВЦКП СО АН СССР	49

Радулов, И.Р., И.Л.Владиков, М.Т.Калчева	
Об одном подходе при включении цифрового терминала ИЗОТ 3500 в системы коллективного пользования	59
Савинков, В.М., О.М.Вейнеров, М.С.Казаров, А.А.Александров	
Обобщенные процедуры логического проектирования баз данных и уточнения инфологической модели: формальный подход	67
Стогний, А.А., А.И.Кондратьев	
О построении математического аппарата для описания процессов проектирования и функционирования информационных систем	83
Терзиев, А.И.	
Управление формата отчетов и поиска информации в базах данных вычислительных систем с множественным доступом	97
Швертнер, И.В.	
Защита корректности данных при вводе в СУБД БИСЕС	105
Швертнер, И.В., Л.Манасиев	
Системный журнал в СУБД БИСЕС	111
Эскенази, А.М., Н.М.Манева, В.Т.Петрова	
Поиск при помощи инвертированных файлов в системе БИСЕС	117

выпуск 133 - том III

Barnev, P., At.Radensky, P.Azalov, Kr.Markov, Z.Vassilev	
A local information station - Version one	7
Benczur, A.	
Problems in modelling of data base performance	27
Bittner, J.	
DBS/R - A system of practice	43

Demetrovics, J., Gy. Gyepesi	
Logical dependencies in Relational	
Data Base	59
Kerékfy, P.	
Some remarks on statistical data	
processing	79
Havel, I., P. Liebl	
A relational DBMS in Concurrent PASCAL	99
Riha, A.	
Modifiable query system for casual Data	
base user	III
Werner, W., D. Koch	
Natural language interfaces to Data	
bases: a day-dream or a realistic goal?	I23

П Р Е Д И С Л О В И Е

Комиссия Научные Вопросы Вычислительной Техники (КНВВТ) является органом, созданным для координации многостороннего сотрудничества между Академиями наук социалистических стран по проблеме "Научные вопросы вычислительной техники". Комиссия была создана согласно решению I Совещания Академий наук социалистических стран, которое состоялось в 1962 году в Варшаве.

В рамках КНВВТ проводятся исследования в следующих направлениях:

1. Вычислительные методы
2. Методы программирования
3. Теоретические основы вычислительной техники
4. Применение математических методов и ЭВМ.

Одной из главных форм сотрудничества являются Рабочие группы (РГ). РГ создаются для разработки особо важных задач по тематике КНВВТ для решения конкретных научно-технических проблем, составления обзоров состояния определенных научных направлений и разработки технических заданий для создания пакетов программ. Как правило РГ создаются сроком на четыре года для выполнения конкретной определенной задачи. Заключительный научный отчет о работе РГ представляется председателем РГ на заседании КНВВТ.

КНВВТ на XV заседании утвердила инициативный комитет по предложению о создании новой рабочей группы в направлении систем управления базами данных и информационных систем. В заседании инициативного комитета приняли участие представители БАН, АН ГДР, АН СССР и ЧСАН.

Создание рабочей группы РГ-11 "Системы управления ба-

зами данных и информационные системы" было утверждено на XVI заседании Комиссии многостороннего сотрудничества между Академиями наук социалистических стран, которое состоялось с 8 по 10 мая 1977 года в Гаване, Куба.

Ответственность для РГ-11 приняла Болгарская АН, как председателем был предложен и утвержден доц. Д. М. Добрев. На первом заседании РГ-11 (с 9 по 13 мая 1978 года, Варна, НРБ) как Сопредседателем был предложен и утвержден старший н.с. Г. К. Столяров (АН СССР), а как Секретарем - канд. мат. наук Р. К. Киркова (БАН). Как представителями сотрудничающих в РГ-11 АН социалистических стран были утверждены: Д. М. Добрев для БАН; А. Бенцур для БАН; Хо Тхуан для ЦННБ; Б. Бельке для АН ГДР; П. Димитет для АН Кубы; Я. Банковски для ПАН; А. А. Стогний для АН СССР; И. Хавел для ЧСАН.

Участники РГ-11 решили включить в план научного сотрудничества следующие направления:

1. Терминология и классификация систем управления базами данных (СУБД) и информационных систем (ИС)
(ответственные Д. М. Добрев, А. А. Стогний, Г. К. Столяров)
2. Разработка рекомендаций по архитектуре, компонентам и интерфейсам в СУБД
(ответственный Д. Шуберт)
3. Методы описания и оценки СУБД
(ответственный Я. Банковски)
4. Автоматизация проектирования информационных систем и банков данных
(ответственный А. Бенцур)

На втором заседании РГ-11 (с 29 по 31 января 1979 г., Вуданешт, БНР) в результате обсуждения было принято решение о расширении тематики РГ-11 новым направлением

5. Базы данных и информационные системы в вычислительном центре коллективного пользования (ВЦКП) и в сети ЭИМ
(ответственный Д. М. Добрев)

Расширение тематики этим направлением считалось целесообразным для проведения некоторых предварительных исследований в связи с подготовкой условий будущих исследований (эventуально вне тематики РГ-11) в области систем управления распределенными базами данных (СУРБД).

По отдельным направлениям работа проводилась в рамках утвержденные РГ-11 национальные и многонациональные коллективы. Эти коллективы предоставляли на утверждение планы своих исследований и отчитывали свои результаты в РГ-11. Для этой цели РГ-11 провела восемь заседаний, большинство из которых были сопутствованы проведением конференциями или сессиями в некоторых из направлениях научных исследований РГ-11. Эти мероприятия были либо организованы, либо стимулированы сотрудниками РГ-11. В конце этого сборника в качестве приложения приводятся: Список мероприятий РГ-11; Имена и адреса участников РГ-11 .

За время свое существования в заседаниях РГ-11 активно участвовали 35 специалистов из восьми АН социалистических стран. Они являются ведущими специалистами сотрудничающих АН в области научных исследований РГ-11. Кроме того в работе РГ-11 принимали активное участие свыше 100 специалистов как в изготовлений отдельных рабочих или отчетных материалов, так и в организованных в рамках РГ-11 мероприятиях.

За время четырехлетнего существования в рамках совместной работы РГ-11 были разработаны ряд материалов, некоторые из них были оформлены как научные публикации, а другие как отчеты в соответствующих научных направлениях.

Участники РГ-11 обращают особое внимание на оформление окончательных результатов исследований имея ввиду, что этими результатами будут пользоваться широко в сотрудничающих стран. С этой целью было принято решение:

1. Опубликовать в отдельном сборнике научно-исследовательские работы, завершённые в рамках РГ-11. Ответственность на опубликование этого сборника приняла на себя БАН.

Этот сборник является первым изданием таких материалов.

2. Разработать итоговый отчет с рекомендациями о дальнейшей работе в области систем управления базами данных (СУБД) и информационных систем (ИС), содержащий в качестве приложения полученные по ходу совместных исследований материалы.

Ряд научных результатов, полученные в рамках сотрудничества по линии РГ-11, были опубликованы в разных ведущих журналах. Считается целесообразным в следующих изданиях сборника привести некоторые из них в переводе на русский язык, если они были изданы на национальном языке.

Сотрудники РГ-11 очень признательны Руководству Исследовательского института вычислительной техники и автоматизации (ИИВТА) Венгерской АН за предоставленную возможность опубликовать этого сборника, а также и редакционной коллегии, которая тщательно отредактировала и отпечатаала материалы этого сборника.

Январь 1982 года

Председатель РГ-11 Д.М.Добрев

ПРОБЛЕМЫ СОЗДАНИЯ ВЫЧИСЛИТЕЛЬНЫХ ЦЕНТРОВ
КОЛЛЕКТИВНОГО ПОЛЬЗОВАНИЯ В КОМИТЕТЕ ПО
Е С С И

А.П.Александров

Комитет по ЕССИ

Единая система социальной информации (ЕССИ) представляет собой общенациональную систему для охвата, обработки, сохранения и предоставления информации, необходимой для управления социально-экономического развития страны.

Единая система социальной информации включает:

- информационные системы функциональных ведомств, как например: планирование, статистика, финансово-кредитная система и др.;
- информационные системы хозяйственных министерств, как например: строительство, транспорт и др.;
- территориальные информационные системы.

Техническая база ЕССИ представляет собой система технических и программных средств сбора, подготовки, передачи, обработки и предоставления информации. Территориальные вычислительные центры являются основными элементами технической базы ЕССИ.

Под вычислительным центром коллективного пользования (ВЦКП) понимаем вычислительный центр, обладающий информационными и вычислительными ресурсами, которые могут предоставляться широкому кругу потребителей через развитой сети коммуникаций (система телеобработки и связь с многомашинной сетью). Из этого следует, что вычислительный центр коллективного пользования является основной составной частью в одной широко развитой системе для распределенной обработки данных.

Исходя из функций и задач Комитета по ЕССИ следует, что территориальные информационно-вычислительные центры (ТИВЦ) должны развиваться как вычислительные центры коллек-

тивного пользования, так как в сущности они и теперь уже являются центрами коллективного пользования, но в самом примитивном виде. Для создания ТИВЦ как ВЦКП необходимо преодолеть следующие проблемы.

1. До сих пор приложение вычислительной техники в округах НРБ шло путем создания ТИВЦ, которые в дальнейшем искали своих клиентов на предприятиях, работающих в режиме локальной пакетной обработки. Этим путем можно автоматизировать и интенсифицировать только последних 20% цикла сбора, обработки и использования информации, и из-за этого работа ТИВЦ оказывается низкоэффективной. Чтобы интенсифицировать всего цикла необходимо охватывать информацию в процессе ее возникновения.

Итак, ТИВЦ-КП нельзя создать только усилиями ТИВЦ и Комитета по ЕССИ, потому что, чтобы ТИВЦ был центром коллективного пользования необходимо на предприятиях округа создать пункты сбора и первичной обработки данных. Создание ТИВЦ-КП должно идти не только в направлении от ТИВЦ к предприятиям, но и обратно. В противном случае ТИВЦ останутся как сердце без кровеносной системы.

Итак, первая проблема создания ВЦКП является стратегической. Эти центры могут создаваться только если их создание превратится в государственную политику, которая координирует действий Комитета по ЕССИ и всех ведомств.

2. Для создания пунктов сбора данных и пунктов сбора и первичной обработки данных нужны серийно производимых типов терминалов и конечных устройств. Пока в социалистическом лагере только ГДР производит технические устройства, которые частично могут ответить на эту необходимость. Эти устройства однако очень специфичны в техническом и программном отношении и поэтому их приложение в нашей стране бесперспективно.

Перспективы развития СМ ЭВМ и ЕС ЭВМ не вполне ясны. Министерство электроники и электротехники начинает разрабатывать различных программно-ориентированных комплексов (ПОК),

предназначенных для передачи клиентам как готовую систему технических и программных средств. С нашей стороны не раз велись разговоры о включение в программу ПОК, необходимый для наших целей. Им даны подробные требования и примерную комплектацию необходимого ПОК-а. Итак, вторая проблема является технической и состоит в том, что нет системы серийно производимых технических средств, при помощи которых можно создавать БЦКП.

3. Пока не существует явных экономических стимулов для внедрения вычислительной техники на предприятиях. Этот фактор приобретает большую тяжесть в условиях нового экономического механизма. Тяжесть этого фактора покажем на следующих примерах.

Стоимость одного часа машинного времени приблизительно равна одной месячной зарплате одного бухгалтера. Это означает, что за одного часа машина должна совершить больше работы, чем совершает бухгалтер за месяц, или что руководитель предприятия во всех случаях предпочтет назначить одного бухгалтера, нежели платить за один час машинного времени. Для сравнения можно сказать, что в некоторых странах стоимость 1 часа машинного времени является $1/10$ до $1/20$ средней месячной зарплаты.

Вопрос о стоимости 1 часа машинного времени связан и с ценой технических средств, которая относительно высока.

Так например: стоимость одного неинтеллигентного видео-терминала на международном рынке колеблется от 1000 до 2500 долларов, что равно приблизительно одной до двух с половиной месячных разплат. Стоимость аналогичного венгерского видео-терминала больше двадцати средних годовых заработков. В таком случае за сколько времени и каким способом откупится использованный терминал?

Картина усложняется тем, что для создания БЦКП необходимо использовать канал для передачи данных. Наем за каналы для одной системы телеобработки за год приблизительно равна стоимости системы, а это означает, что стоимость эксплуатации этой системы за год равна капиталовложениям при ее создании. Комитет по ЕССИ делал много раз усилия изменить необоснованно высокие цены за каналы и корректировать явные несообразности в тарифах, но до сих пор без существенного результата.

Итак, третья проблема является экономической и относится не только к экономической эффективности ВЦП, но и к экономической эффективности от внедрения электронной обработки данных.

Пока общего решения поставленных проблем нет. В некоторых конкретных случаях эти проблемы находят свое решение. Пути решения этих проблем видны только в реальной практической работе для постепенного создания в ТИВЦ необходимых информационных и технических ресурсов для их превращения в ВЦП.

ПРОБЛЕМЫ УПРАВЛЕНИЯ В СИСТЕМАХ ИНФОРМАЦИОННОГО ОБСЛУЖИВАНИЯ КОЛЛЕКТИВОВ

П. Барнев, Кр. Марков

Институт математики Болгарской АН, София, НРБ

1. Введение.

Настоящая работа посвящена некоторым проблемам, связанным с системой управления Абстрактной системы информационного обслуживания коллективов (СИОК) [1, 2, 3, 4, 5, 6, 7, 8].

Напомним, что СИОК предназначена для обслуживания небольших коллективов, с численностью несколько десятков человек, имеющие очень развитую информационную деятельность, как, например, научные или проектантские коллективы.

Абстрактная СИОК состоит из двух компонентов – архив и процессор. Общая схема системы и ее взаимодействия с коллективом показаны на рис. 1. Процессор СИОК составлен из трех систем – система обмена информации (СОИ), система обработки (СО) и система управления (СУ). Далее рассматривается система управления.

Разработка системы управления СИОК имеет экспериментальный характер. Цель эксперимента – проверить на практике некоторые идеи и методы управления, которые используются в других областях человеческой деятельности, но до сих пор не применялись в вычислительных системах.

2. Характерные черты коллективов

В СИОК коллективы рассматриваются с точки зрения их информационной деятельности. Считается, что они отличаются:

- определенностью их целей и задач, позволяющих оценить эффективность работы каждого члена коллектива;
- хорошим планированием, позволяющим совместить и синхронизировать деятельности отдельных членов коллектива;
- отчетливой внутренней структурой;
- регламентированностью деятельности, прав и обязан-

ностей членов коллектива и наличием хороших систем контроля, отчета и борьбы с нарушениями трудовой дисциплины.

3. Принцип управления СИОК

Развитие управления вычислительных систем (ВС) можно рассматривать в двух периодах:

1) Управление ВС человеком

Как известно, этот период начался с прямым управлением ВС каждым потребителем. Вследствие конкурентного использования ВС, между потребителями появились противоречия. Постепенно работа на машине была передана человеку - оператору. Развились наборы вспомогательных программ, облегчающие его деятельность. Дальше, эти наборы переходят в операционные системы ВС. Так достигнули второй период.

2) ВС сама управляет свою работу.

На переднем плане была выдвинута эффективность ВС. Операционные системы создавались так, чтобы они управляли работой ВС таким образом, что организация выполнения потребительских заданий (или их частей) была бы в высшей степени выгодна для ВС. Противоречия между потребителями разрешались с точки зрения эффективности ВС. Таким образом, ВС постепенно начали терять отличительную черту обслуживающих систем - вместо того, чтобы они подчинялись потребителям, ВС стали ими управлять.

Отметим, что характерно для этого периода является то, что управление ВС - оперативное и базирующееся только на текущих ситуациях. Отсутствует долгосрочное управление работы ВС.

Основным принципом в рассматриваемой СИОК является то, что система вновь приобретает подчиненную роль. Однако, это подчинение происходит на новом уровне - автоматизированная система информационного обслуживания подчиняется не отдельному члену коллектива, а коллективу в целом, представляемому его руководством. Коллектив воздействует на систему через свое руководство (рис. 2). В рамках коллектива, противоречия между потребителями или почти не возникают, или решаются

руководством с точки зрения целей и задач всего коллектива.

4. Структура управления в СИОК

Управление в СИОК строится иерархическим образом и распределяется на нескольких уровнях (рис. 3):

- руководство коллектива (высший уровень);
- главная управляющая система (ГУС) - первый уровень СУ СИОК;
- управляющие программы (УП) отдельных систем процессора СИОК - для обмена информации, для обработки и для управления - второй уровень СУСИОК.

5. Функции системы управления СИОК

Система управления СИОК имеет две основные функции:

- управлять всеми внутренними работами СИОК;
- обслуживать управленческую деятельность руководства коллектива.

Одна из тенденций развития ВС и их математического обеспечения является постепенная децентрализация функций управления. В сравнении с централизованной структурой, децентрализованная имеет свои преимущества и недостатки. Преимущества являются например, автономность отдельных компонентов системы, гибкость их замены, удобство при переходе к многомашинным комплексам. К недостаткам можно причислить дублирование некоторых функций системы, определенный перерасход ресурсов, усложнение координации работ самостоятельно управляемых подсистем.

В системе управления СИОК реализуется централизованно-децентрализованная функциональная структура. Точнее в СУ СИОК:

- централизируются функции управления, связанные с обслуживанием долгосрочного управления коллективом
- децентрализируются функции оперативного управления СИОК, с целью создания удобной экспериментальной модели для исследования проблем связанных с параллельным обслуживанием, с переходом от одномашинных к многомашинным ВС и др.

6. Обслуживание управленческой деятельности руководства коллектива.

Управление коллективом тоже является информационной деятельностью и следовательно можно искать способы автоматизации его обслуживания.

Управление коллективом главным образом осуществляется в двух направлениях:

1) Управление информационными деятельностями членов коллектива, направленными на достижение его основных целей.

2) Управление вспомогательными деятельностями в коллективе, такие, как финансовая отчетность (напр. расчет зарплат), материально-техническое снабжение и т.д.

В СУ СИОК все усилия направлены на обслуживание первого направления, так как деятельность во втором направлении не является основной и не имеет большое значение для немногочисленных коллективов. Это обслуживание является основной задачей Главной управляющей системы. Все перечисленные ниже функции выполняются ею.

Управление информационными деятельностями членов коллектива в основном является долгосрочным управлением. Однако, некоторые его функции - это функции оперативного характера.

Обслуживание функций долгосрочного управления выражается в прослеживании и контроле соблюдения и выполнения долгосрочных планов работы коллектива, в накоплении соответствующей информации и выдачи сигнальной информации, как правило, в заранее определенные моменты суток, недели, месяца. Сигнальная информация указывает, например, что:

- некоторые члены коллектива не выполняют свои обязанности, такие, как соблюдение графика внесения информации в архив системы,

- нарушены принятые критерии оценки эффективности работы коллектива или СИОК.

При желании, руководство может использовать и накопленную информацию. Принимая решения, соответствующими директивами оно может направлять работу системы в нужном направлении, меняя, например, приоритет некоторых деятельностей или план распределения ресурсов системы и коллектива. Директива-

ми можно задавать и изменения в составе и структуре коллектива, обновление планов и т.д.

Обслуживание этих функций руководства, имеющих оперативный характер, выражается в восприятии оперативных директив руководства и в выполнении соответствующих реакций.

Отметим, что ГУС руководит и работой, связанной с контролем соблюдения дисциплины взаимодействия коллектива с СИОК.

7. Управление внутренних работ СИОК

Это управление является в основном оперативным и децентрализованным. Оно подчиняется планам работы коллектива и координирует обслуживание параллельно протекающих информационных деятельностей членов коллектива.

Об оперативном управлении в СИОК заботятся управляющие программы второго уровня (УП систем обмена информации, обработки и управления). Они, локально, в рамках своих систем, распределяют предоставленные им ресурсы, контролируют и распределяют работу между частями соответствующих систем и заботятся о повышении эффективности работы этих систем.

Так, как прямая коммуникация между систем процессора СИОК не разрешена, одна из функций Главной управляющей системы (первого уровня) является обеспечение и управление этой коммуникацией, как и глобальное распределение ресурсов СИОК между этими системами.

Все сервисные функции внутреннего управления СИОК сосредоточены в отдельном блоке системы управления, формирующем своеобразный третий уровень управления в СИОК. В этом блоке находят место такие функции, как:

- преобразование, сохранение, обмен и обновление служебной для СИОК информации,
- анализ потребления и перераспределение ресурсов,
- принятие мер для защиты СИОК от неправильной или аварийной работы,
- управление параллельного обслуживания поступающих от коллектива запросов.

Заключение

Абстрактная СИОК будет реализована в нескольких этапах. В настоящем времени реализуется первый этап, который будет развиваться и расширяться. Однако, еще на этом этапе особое внимание уделяется СУ и специально, долгосрочному управлению и соответствующему контролю. Предусматривается и децентрализация управления в СИОК.

Дальнейшее развитие системы управления требует решение ряд других вопросов, которые не были затронуты в настоящей работе. Такими вопросами являются, например, управленческие вопросы, связанные с активным сбором и распространением информации.

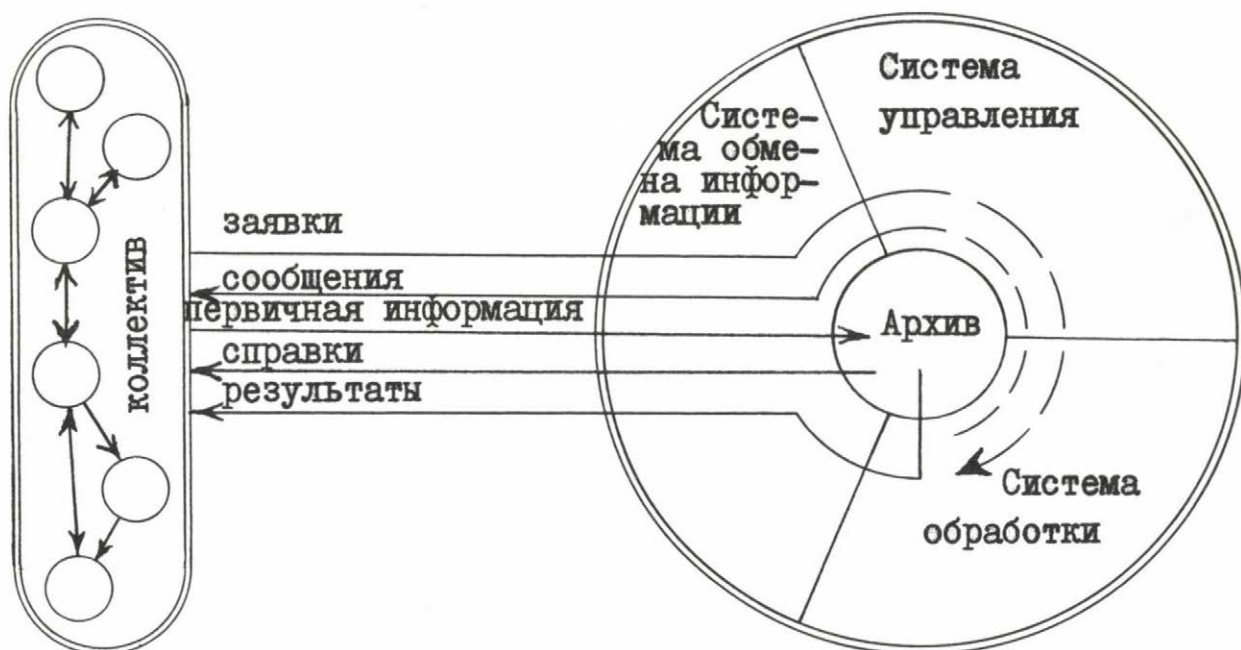


Рис. 1

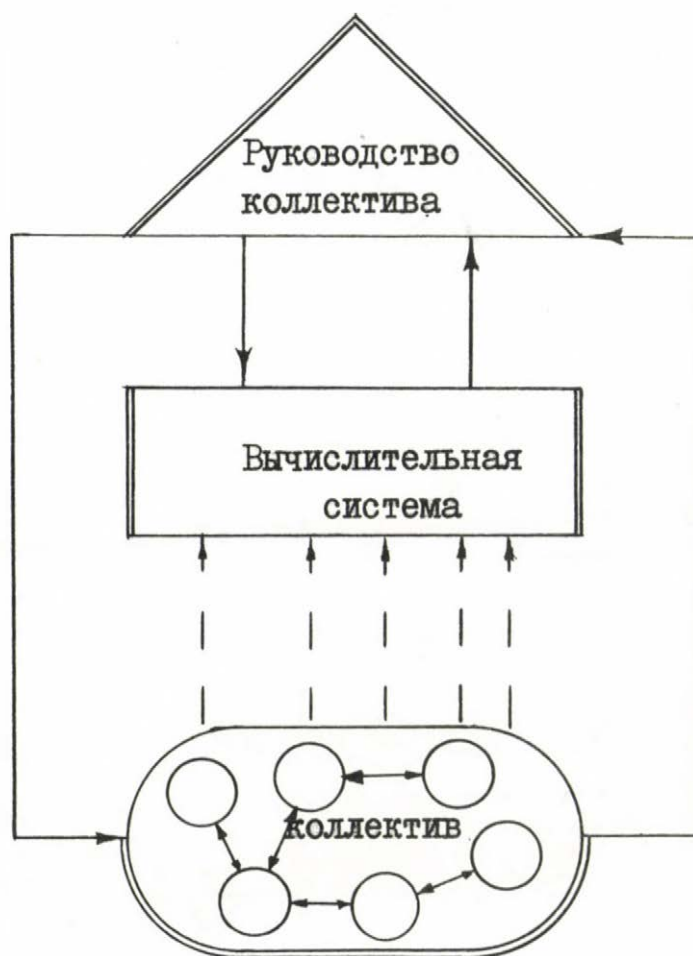


Рис. 2

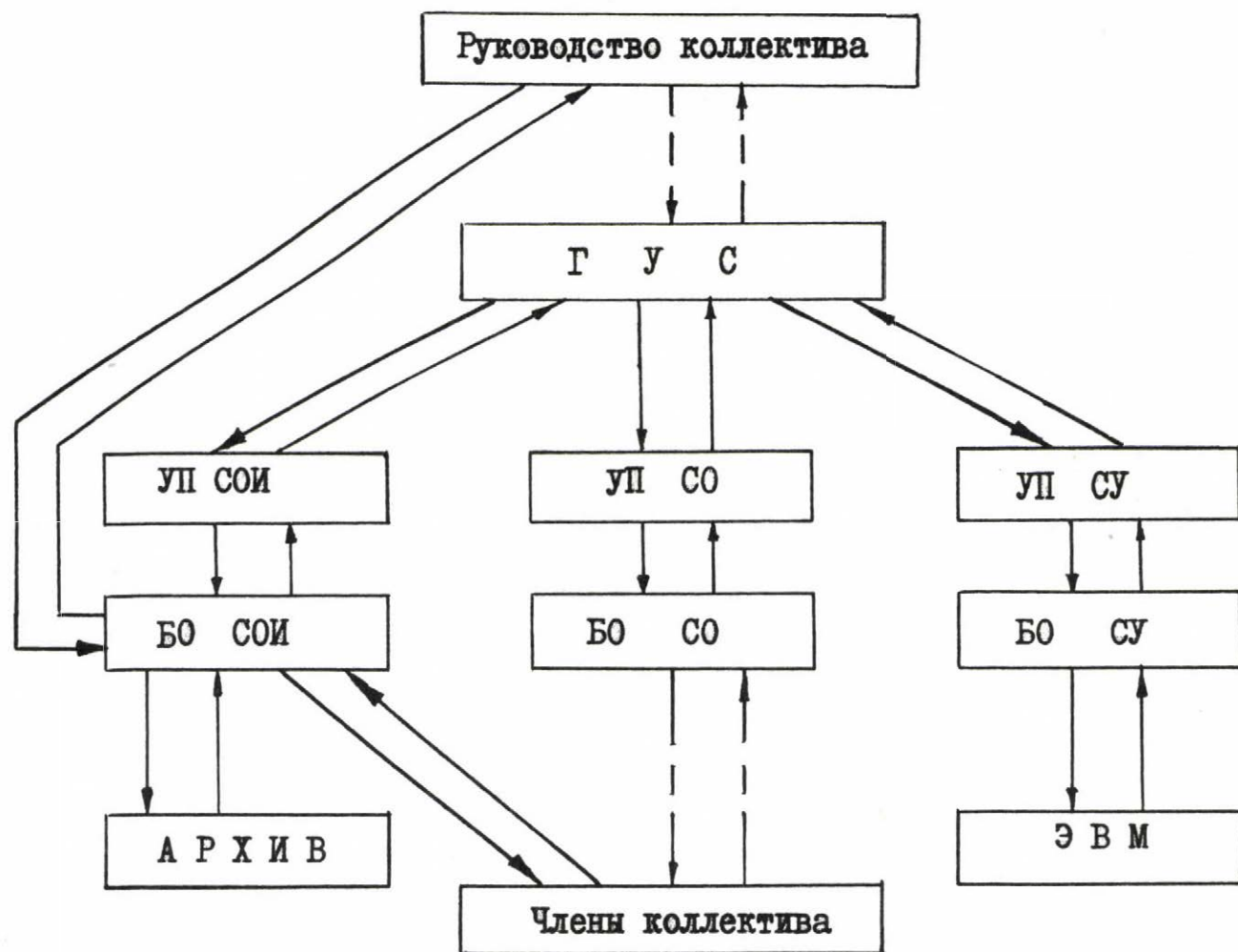


Рис. 3

Л и т е р а т у р а

1. P. Barnev, Systeme abstract de service informatique des collectivites, Congres de mathematiques appliquees, Thessalonique, 1976.
2. P. Barnev, Systems for information servicing of collectivities, Serdica, Vol. 4, fasc. 2-3, 1978.
3. P. Barnev, A. Radenski, Structure of the information and operations on the entities in a system for information servicing of collectivities, Serdica, vol. 4, fasc. 2-3, 1978.
4. K. Ivanov, Concretization of the collectivity and individualizing the Communication Language in a System for Information Servicing of Collectivities, Serdica, vol. 4, fasc. 2-3, 1978.
5. П. Бърнев, К. Иванов, Входно-изходна система в системата за информационно обслужване на колективи, IV Международен симпозиум по передачи данных и обработке информации, 26-31 мая 1977, Варна.
6. П. Бърнев, Вл. Занев, А. Раденски, Бази от данни. Система за информационно обслужване на колективи, 8 пролетна конференция на СМБ, Сл. бряг, април 1979.
7. П. Бърнев, А. Раденски, Вл. Занев, Структури данных в архиве системы информационного обслуживания коллективов, Сборник докладов "Конференция по приложению вычислительной техники и баз данных в научных исследованиях", Варна, май 1978
8. Вл. Занев, Концепция и модель распределенной системы информационного обслуживания коллективов. В настоящем сборнике.

ИМИТАЦИЯ РАБОТЫ СЕТЕЙ ЭВМ

Л.Н.Батурина, Н.А.Лепешинский

Белгосуниверситет имени В.И.Ленина, г. Минск, СССР

Одним из перспективных направлений использования вычислительной техники является создание многоуровневых вычислительных систем или сетей ЭВМ.

Информационно-вычислительные сети относятся к сложным системам, характеризующимся наличием большого числа разнородных элементов и связей между ними. Поэтому на этапе проектирования сети при решении задач выбора ее структуры целесообразно использовать выводы общей теории систем, которые, в частности, подчеркивают определяющую роль организации взаимодействия между подсистемами, направленной на согласование (координацию) их работы с целью достижения оптимального значения некоторого критерия функционирования системы. Как показывают теоретические исследования в области сложных систем [1,2] и опыт создания конкретных сетей ЭВМ [3,4], наиболее целесообразной формой организации таких систем является многоуровневая иерархическая структура, в которой на основе обмена информацией между уровнями обеспечивается возможность согласования подсистемных (локальных) критериев оптимальности с глобальными при относительной свободе локальных действий [1,5]. Ниже для решения задачи выбора оптимальной иерархической структуры основанной на сети ЭВМ, предложен метод оптимального моделирования.

I. В задаче выбора структуры сети ЭВМ требуется указать конкретные узлы сети и связи между ними, так чтобы определенный критерий эффективности достигал оптимального значения при заданных граничных условиях. Критерий эффективности должен характеризовать исследуемую систему как единое целое и обеспечивать количественную оценку качества функционирования системы с требуемой точностью. Для поставленной задачи этим

требованиям удовлетворяет, например, такой критерий, как среднее время пребывания задания в сети. Эта величина включает в себя время поиска в сети необходимых вычислительных ресурсов, время, необходимое для передачи к ним задания в очередь на выполнение, время пребывания в очереди и передачи готового решения пользователю..

Значения выбранного критерия зависят от таких параметров сети, как интенсивность входного потока заданий, производительность узлов сети, пропускная способность линий связи, организация связей между узлами сети, организация управления работой сети. С точки зрения построения математической модели существенно, что два последние названные вида параметров являются качественными и не представляется возможным построить аналитическую зависимость критерия эффективности от способа организации связей между узлами сети и управления ее работой.

Именно эта особенность послужила обоснованием для выбора имитационного моделирования в качестве основного метода исследований.

2. Объектом исследования явилась имитационная модель гипотетической многоуровневой сети ЭВМ с тремя уровнями иерархии. Третий (высший) уровень представлен одной ЭВМ наибольшего быстродействия τ_3 , которая выполняет функцию координатора работы всех подуровней сети. Второй уровень иерархии включает ЭВМ быстродействия $\tau_2 < \tau_3$ каждая из которых связана с ЭВМ быстродействия $\tau_1 < \tau_2$ низшего (первого) уровня иерархии. Вычислительные машины второго уровня выполняют функции координатора работы связанных с ними ЭВМ первого уровня.

Будем считать, что входной поток заданий, поступающий с терминальных устройств, распределяется так, что 60% заданий поступают на ЭВМ первого уровня, 30% заданий – на ЭВМ второго уровня, 10% заданий – на ЭВМ третьего уровня. Для моделирования работы сети необходимы следующие данные:

- интенсивность потока заданий;
- закон поступления заданий в сеть;

- максимально возможное время T_i пребывания i -го задания в сети;
- время T_{ik} обработки i -го задания на ЭВМ типа k ($k = 1, 2, 3$).

Кроме того, каждый узел сети характеризуется быстродействием γ_k ($k = 1, 2, 3$) и заданным максимальным уровнем мультипрограммирования. Пропускная способность всех каналов связи считается одинаковой.

Основу метода имитации составляет следующий общий алгоритм, описывающий функционирование сети при заданной организации связей ЭВМ и управления ее работой. Алгоритм состоит из следующих этапов.

1) Для i -го задания, поступившего на ℓ -ю ЭВМ n -го уровня, анализируется возможность его выполнения на данной ЭВМ, т.е. осуществляется проверка условия

$$T_i \geq \sum_{j=1}^{m+1} t_{jk} + t_{ik} + t_e^{отв} \quad (I)$$

где m - длина очереди заданий на обработку на ℓ -ой ЭВМ типа k ;

$t_{m+1,k}$ - время, в течение которого будет выполняться задание, занимающее процессор ℓ -ой ЭВМ в момент поступления i -го задания;

t_{ik} - время, необходимое на выполнение i -го задания на ℓ -ой ЭВМ типа k ;

$t_e^{отв}$ - время, необходимое для передачи готового решения с ℓ -ой ЭВМ абоненту.

2) Если для i -го задания условие (I) выполняется, задание ставится в очередь к ℓ -ой ЭВМ на обработку.

3) Если условие (I) не выполняется, то при $n < 3$ связанная с данной ℓ -ой ЭВМ вычислительная машина уровня $n+1$ анализирует возможность обработки задания или на основе собственных ресурсов, или на основе ресурсов других ЭВМ более низкого уровня, работа которых находится под ее контролем. Это сводится к проверке выполнения условия:

$$T_i \geq \sum_{j=1}^{m+1} t_{jk} + t_{ik} + t_{se}^{отв} + t_{es}^n + t^p \quad (2)$$

где $T_{se}^{отв}$ - время, необходимое на передачу результатов обработки i -го задания с S -ой на e -ю ЭВМ (куда заявка поступила первоначально) и затем обоненту;

t_{es}^n - время, необходимое для передачи задания на выполнение с e -ой ЭВМ на S -ю;

t^p - время, затрачиваемое координатором на поиск необходимых вычислительных ресурсов.

4) Если $n=3$ и условие (1) для i -ой заявки не выполняется, ЭВМ третьего уровня анализирует сначала возможность обработки задания на ЭВМ уровня $n-1$ по условию (2). Если необходимая ЭВМ найдена, задание передайте к ней на обработку. В обратном случае проверяется выполнение условия (2) для ЭВМ уровня $n-2$. Если необходимая ЭВМ найдена, задание передается к ней на обработку. При отсутствии необходимых вычислительных ресурсов задание покидает систему не обработанным.

5) Если условие (2), проверяемое в п.3 не выполняется на уровне $n+1$, то к поиску необходимых ресурсов подключается ЭВМ третьего уровня и повторяется п.4.

В ходе работы алгоритма осуществляется сбор статистических данных по числу заданий, которые покинули сеть необслуженными.

3. При реализации имитационной модели на основе описанного выше общего алгоритма, как и при реализации любой модели широкого класса, необходимо решить следующие основные задачи [6] :

1) организовать динамическое распределение памяти;

2) организовать псевдопараллельное выполнение алгоритмов, описывающих функционирование отдельных подсистем модели;

3) включить в модель возможность имитации случайных процессов;

4) реализовать необходимые дисциплины обслуживания заявок;

5) осуществить сбор и обработку результатов моделирования.

Эффективное решение всех этих задач можно осуществить, используя специализированные языки моделирования. В качестве такого языка был выбран язык ПЛИС, предназначенный для построения программных имитационных моделей дискретных систем [7] .

4. Проведение эксперимента на имитационной модели гипотетической сети ЭВМ, позволяет изучить влияние выбранной структуры соединения узлов сети и управления работой сети на значение заданного критерия эффективности, на величины коэффициентов использования каждого элемента сети, а также на длины очередей заданий при обслуживании их каждым элементом сети, т.е. на пропускную способность исследуемой сети .

Так, например, исследования сети, в которой допускается наличие каналов связи только между ЭВМ различных уровней, показали, что такая структура не позволяет обеспечить работу при определенных, практически обоснованных значениях параметра T_i -максимально возможного времени нахождения каждого i -го задания в сети. Ограниченное количество линий связи при данной структуре приводит к перегрузке каналов, так как передача в случае необходимости заданий между ЭВМ одного уровня осуществляется лишь через более высокий уровень. При этом выход из строя любого канала связи сразу прерывает связь между ЭВМ двух уровней и таким образом нарушается возможность координации работы подсистем более низкого уровня. Это приводит к значительному увеличению времени пребывания задания в сети.

Выявленные недостатки строго иерархической структуры связи изучаемой сети требуют проведения дальнейших исследований в направлении поиска оптимальной структуры, обеспечивающей надежность работы сети за счет введения в структуру дополнительных каналов связи. Необходимость введения дополнительных линий связи в сети может возникнуть также при решении вопросов уменьшения нагрузки на каналы, соединяющие ЭВМ разных уровней. Кроме того, вопросы обеспечения надежности работы вычислительной системы возникают при выходе из строя ЭВМ любого уровня, так как удлиняется время пребывания задания в сети и может нарушаться структура управления.

Исследование перечисленных задач приводит к необходимости внесения корректив в общий алгоритм функционирования модели вычислительной системы. Например, при введении дополнительных каналов связи возникает необходимость определения кратчайшего пути передачи задания другим ЭВМ с целью выполнения ограничения по T_i . Для этого в программу имитационной модели вводится процедура определения величин t_{se}^{orb} и t_{se}^n .

Литература:

1. Месарович М., Мако Д., Такахара Я. Теория иерархических многоуровневых систем. М. "Мир", 1973.
2. Бусленко Н.П. и др. Лекции по теории сложных систем. М. "Сов.радио". 1973.
3. Какунин Л.А. Проблемы проектирования и разработки вычислительных сетей. "Заруб. радиоэлектроника", № 9, 1978.
4. Boorstyn Robert R., Frank Howard. Large-scale network topological optimization. "IEEE Trans. Commun." 25 №1 1977
5. Кухтенко А.И. О теории сложных систем с иерархической структурой управления. Сб. "Сложные системы управления", Киев, 1966.
6. Основы теории вычислительных систем. М. "Высшая школа", 1978.
7. Система имитационного моделирования ПЛИС. Описание языка. Москва, МИЭМ, 1978.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА КЛАССИФИКАЦИИ СУЕД

Бельке Вольфганг, Хартманн Ханс-Д., Лойхт Бэрбель
Технический Университет Дрезден-Секция обработки информации

1. Введение

На секции "Обработка информации" Дрезденского Технического университета рабочий коллектив занимался проблемами классификации СУЕД с помощью ЭВМ.

При этом было надо выполнить следующие задачи:

- разработка классификатора, описывающего СУЕД,
- развитие и тестирование программ для запоминания, поиска и выдачи информации и
- тест классификатора известных систем.

Целью работы была поддержка консультационного центра для баз данных, где интересующиеся Системами управления базами данных могут получить информацию о существующих системах. Кроме того, должны быть получены дальнейшие познания об устройстве и областях применения будущих СУЕД.

Основой развития данного классификатора явились как теоретические разработки в области баз данных и способов классификации /KLIMESCH u.a. 1971/, /WEDEKIND 1974/, /BELKE 1979/, так и классификаторы разных разработчиков, например Шуберта /SCHUBERT 1977/, Хугенберга и других /HUGENBERG u.a. 1972/, "Общества математики и обработки данных" в Гонне /AUT 1972/ и работа "Основные характеристики отечественных СУЕД и ИПС" /АВТ 1980/. При этом граничное условие заключалось в полном включении результатов последней работы.

2. Инфологическая структура описывающих признаков

Исходным пунктом в развитии анкеты была инфологическая структура описывающих признаков. Сначала надо было исследовать параметры, которыми интересуются будущие пользователи СУЕД. Из требуемого логического развития признаков получилась древовидная структура. На рис. 1 изображена коренная часть инфологической структуры.

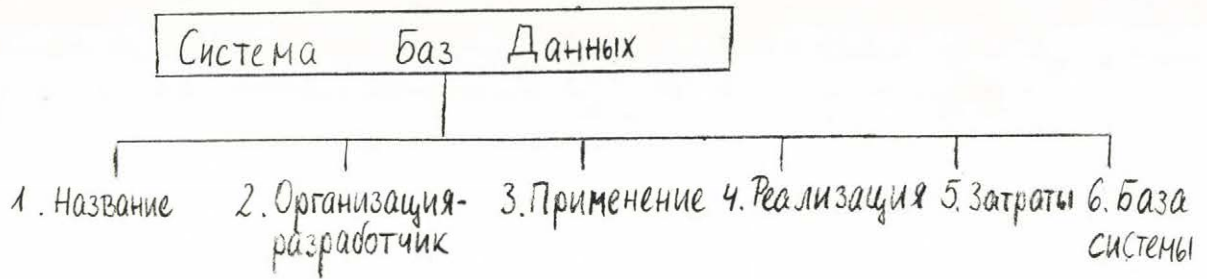


Рис. 1: Коренная часть инфологической структуры

Группами признаков являются:

1. название,
2. организация-разработчик,
3. применение,
4. реализация,
5. затраты и
6. база системы.

Эти шесть групп можно рассматривать как частичные классификаторы общего классификатора.

На рис. 2 частично показана инфологическая структура частичного классификатора "база системы". Концевые узлы представляют собой признаки.

Появляются два вида множеств значений признаков:

1. Значения признака являются свободно избираемыми /открытое множество значений/. Этот вид имеет смысл для большого числа признаков, так как часто нет возможности определить все значения признаков с самого начала.

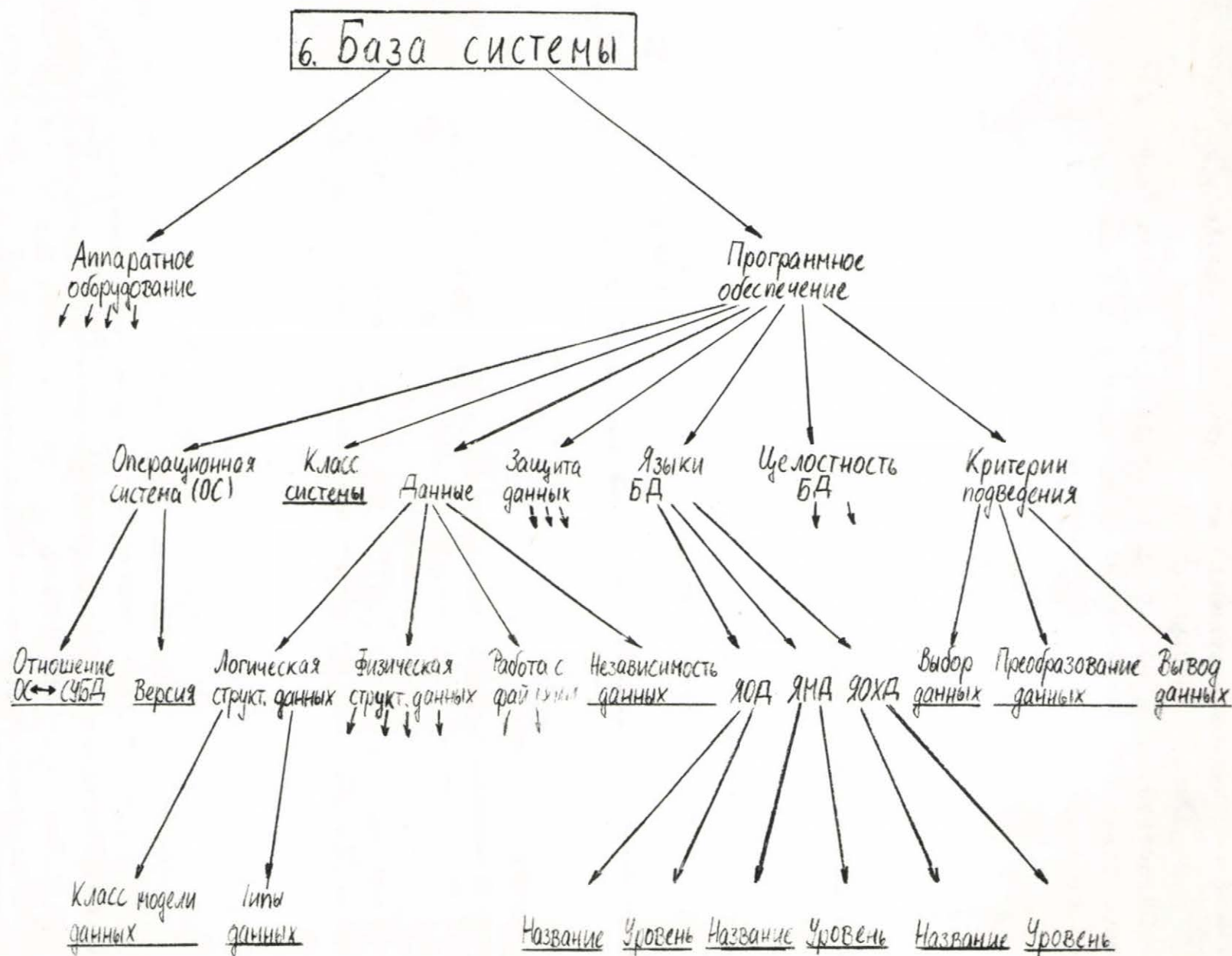
Пример: признак: "Полное название СУБД"

2. Множество значений признаков задано /закрытое множество значений/.

Примеры: -признак: "Уровень" каждого из трёх языков баз данных
значения признака: внутренний, концептуальный, внешний уровень

-признак: "Класс модели данных"

Рис.2: Часть иерархической структуры частного классификатора "База системы"



значения признака: иерархическая, сетевая, реляционная, бинарная модель.

Существует конечно и в этом случае возможность расширения множества значений. Это позволяет включение новых результатов исследования в классификатор. Кроме того, инфологическая структура и тем самым общий классификатор могут быть расширены включением новых свойств.

3. Реализация на ЭВМ

3.1. Общие сведения

При реализации данного классификатора СУБД на ЭВМ был применён язык программирования ПЛ/1 /включительно Макро-ПЛ/1 /, так как он является многообразно применяемым и нужный компилятор имеется в распоряжении в большинстве вычислительных центров.

Этот классификатор используется коллективом разработчиков. Эксплуатация возможна как в ДОС/ЕС, так и в ОС/ЕС на ЭВМ ЕСЕР с ЕС 1022. Для применения используются семь файлов. Каждый из частичных классификаторов занимает один файл. Седьмой файл необходим для запоминания систематики. Все файлы организованы с прямым доступом. Блок "нуль" каждого файла содержит информацию о состоянии файла, например, о числе хранимых описаний СУБД.

3.2. Ввод и запоминание данных

Были созданы программы для запоминания описаний СУБД. С помощью процедур в Макро-ПЛ/1 осуществляется объявление, т.е. описание файлов и структур всех частичных классификаторов. Этим достигается ограниченная независимость данных в системе классификации. Элементы структур, соответствующие признакам, декларируются как строки символов. Это имеет смысл, так как в большинстве случаев значения признаков являются познаковыми данными. В интересах единого представления числовые данные также обрабатываются как строки символов.

Ввод описаний СУБД осуществляется посредством стандартного вводного файла ПЛ/1. Он возможен во всех средах запоминания, допускающие последовательные файлы. Необходимо представить в распоряжении классификаторной программы значения соответст-

венно соглашениям ввода. Существуют три возможности запоминания описаний СУЕД:

1. наполнять отдельно каждый /соответствующий одному частичному классификатору/ файл,
2. запомнить полное описание одной СУЕД,
3. хранить полные описания некоторых СУЕД одновременно.

В интересах соответствия эквивалента "описание системы" - "номер ключа" во всех файлах рекомендуется применение этих двух последних возможностей.

3.3. Выбор и вывод данных

Существуют программы для выдачи описаний. Выдача происходит разрешением древовидной структуры с помощью макро-интерпретатора.

Из этого следуют следующие возможности работы:

- выдача полных описаний СУЕД,
- выдача выбранных значений признаков и групп значений,
- поиск определённых значений признаков.

Любые комбинации этих функций позволяют проводить все необходимые для консультационного центра ЕД операции.

В качестве сред вывода могут быть использованы все среды файла стандартной выдачи. При выдаче на печатающие устройства оформление листов является управляемым при помощи параметров.

На рис. 3 изображено разделение программ системы классификации.

4. Примеры

Рис. 4 покажет отрывок данного классификатора. Часть описания СУЕД "DBS/R" изображена на рис. 5.

5. Литература

/АВТ 1980/: Калинин, Л.А., Кондратьев, А.И., Столяров, Г.К.
Основные характеристики отечественных СУЕД и ИПС.
ИК АН УССР, препринт-80-49, Киев 1980.



Рис. 3: Схема программной системы классификации

- /AUT 1972/: Autorenkollektiv; Kurzbeschreibung von Information-Storage and Retrieval-Systemen. Ges. f. Mathematik u. Datenverarbeitung mbH (Bonn)(GMD), Abteilung für behördliche DV-Systeme, Schloß Birlinghofen 1972.
- /BELKE 1979/: Belke, W., Graichen, D., Starruss, M.; Nichtmetrische Klassifizierung von Informationen. Akademie-Verlag, Berlin 1979.
- /HUGENBERG u.a. 1972/: Hugenberg, D., Kaiser, E.-U.; Beurteilungskriterien für den Einsatz von Datenbank-Management-Systemen in entscheidungsorientierten Informationssystemen. BIFOA-Arbeitsbericht 11/1972, Wison-Verlag, Köln 1972.
- /KLIMESCH u.a. 1971/: Klimesch, H., Lutz, T.; Die Datenbank im Informationssystem, München/Berlin 1971.
- /SCHUBERT 1977/: Schubert, D.; Stand und Perspektiven der Entwicklung von Datenbankbetriebssystemen. RT/DV 14(1977) 4. Beiheft.
- /WEDEKIND 1974/: Wedekind, H.; Datenbanksysteme I. Bibliographisches Institut Mannheim/Wien/Zürich 1974.

6.	SYSTEMBASIS
6.1.	HARDWARE
6.1.1.	KONFIGURATION DER TECHNISCHEN ANLAGEN (MINDESTAUSSTATTUNG)
6.1.1.2.	HS-KAPAZITAET
6.1.3.	PERIPHERE GERAETE
6.1.3.1.	E/A-GERAETE
6.1.3.2.	KAPAZITAET DER PERIPHEREN SPEICHER
6.1.4.	DATEIFERNVERTRAGUNG
6.2.	SOFTWARE
6.2.1.	SYSTEMKLASSE
6.2.2.	BETRIEBSSYSTEM
6.2.2.1.	VERSION BZW. FREIGABE
6.2.2.2.	RELATION ANLAGENBETRIEBSSYSTEM - DBBS
6.2.2.3.	SERVICEPROGRAMME DES DBBS
6.2.3.	DATEIBANKSPRACHE
6.2.3.1.	DBL
6.2.3.1.1.	NAMEN DER DBL
6.2.3.1.2.	SPRACHNIVEAU

- 1. BEZEICHNUNG
 - 1.1. SYSTEM
 - 1.1.1. VOLLSTAENDIGER NAME
DATENBANKSYSTEM/ROBOTRON
 - 1.1.2. ABKUERZUNG
DBS/R
 - 1.2. STADIUM DER AUSARBEITUNG
VERSION 3.0
- 2. HERSTELLER
 - 2.1. KONTAKTPERSON (ANSCHRIFT)
VEB ROBOTRON ZFT, 801 DRESDEN, LENINGRADER STR. 15
 - 2.2. LIEFERBEDINGUNGEN
KA
 - 2.3. DOKUMENTATION
 - 2.3.1. BEZEICHNUNG DER DOKUMENTATION (LITERATUR)
ANWENDUNGSBESCHREIBUNG
 - 2.3.2. UMFANG DER DOKUMENTATION (SEITENANZAHL)
600
- 3. ANWENDUNG
 - 3.1. NUTZERKLASSEN
ADMINISTRATOR, ANWENDUNGSPROGR.
 - 3.2. HAUPTSAECHTLICHES ANWENDUNGSGEBIET DER NUTZUNG
AR-RECHNUNG, STUECKLISTENPROBLEME
 - 3.3. VERWEISE AUF ANWENDER
KA

Рис. 5: Описание СУБА "DBS/R" (отрывок)

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ПАРАЛЛЕЛЬНЫЕ ПРОЦЕССОРЫ С СЕТЕВЫМ ПРОГРАММИРОВАНИЕМ

К.Л.Боянов, В.С.Гетов, Х.А.Турлаков
Институт математики с ВЦ - БАН, София, Болгария
Институт вычислительной техники, София, Болгария

Увеличение скорости обработки данных является одной из основных тенденций развития вычислительной техники. Существуют два основных метода решения этой проблемы - интенсивный, при котором используются более быстродействующие конструктивные элементы и экстенсивный, при котором применяются новые архитектурные принципы. Среди экстенсивных методов увеличения производительности процессоров с последовательным программированием /фон Нейманского типа/ заслуживают внимание магистральный метод /pipelining / [12], предварительный анализ инструкций /look-ahead / [10], матричная обработка [6] и др. При этом, степень параллелизма которая получается в некоторых случаях, на много меньше естественного параллелизма решаемой задачи. Как возможное решение этой проблемы в 60-ых годах родилась идея асинхронного программирования [3]. Его разновидностью является сетевое /data-flow / программирование [5], для которого характерно то, что выполнение каждого оператора программы разрешается после получения его операндов. Используя принцип сетевого программирования можно повысить степень параллелизма до уровня естественного параллелизма решаемой задачи.

Полученные результаты при построении модели процесса асинхронных вычислений являются основой создания процессоров нового типа - сетевых процессоров, для которых характерно параллельное и независимое выполнение операторов данной программы. Определенные возможности в этом направле-

нии дает новая технологическая база и особенно микропроцессоры. В связи с этим, проблемы проектирования сетевых процессоров становятся еще более актуальными. Такие процессоры могут быть как мощными универсальными вычислителями, которые обслуживают одновременно большие коллективы пользователей, так и специализированными или работающими в реальном масштабе времени системами и т.д. В настоящий момент известно несколько разработок сетевых процессоров, которые находятся все еще на этапе эскизных проектов.

Цель предлагаемой работы состоит в формулировании основных требований к сетевым процессорам и обзоре известных архитектурных моделей. На этой основе будут рассмотрены преимущества и недостатки отдельных проектов и определен самый перспективный из них.

I. Прежде чем приступить к рассмотрению конкретных проектов, остановимся на основных требованиях к сетевым процессорам, которые определяют и их характерные свойства.

1. Параллелизм на уровне оператора. Это требование означает возможность выполнения в данный момент времени всех операторов программы, операнды которых имеются в наличии. Выполнение этого требования приводит к увеличению производительности сетевых процессоров.

2. Асинхронность вычислений. В отличие от классических ЭВМ, в которых вычислительный процесс проходит через четко фиксированные состояния памяти, в сетевом процессоре, в зависимости от наличных ресурсов, для конкретной программы реализуются разные, но эквивалентные вычислительные процессы.

3. Модульность. Процессоры с сетевым программированием строятся из типовых модулей, посредством изменения количества и состава которых достигается гибкость как по функциональному отношению, так и в отношении производитель-

ности. Это обеспечивает более высокую надежность и готовность аппаратуры, которые имеют особое значение для современных ЭВМ.

4. Стандартизация межресурсных связей. Она осуществляется посредством коммутации информационных пакетов между компонентами структуры сетевого процессора [9].

Использование асинхронных протоколов устраняет потери времени при обмене между устройствами процессора. Одновременно с этим интерфейс между ними становится более простым.

5. Адаптивность. Это свойство [2] выражает способность сетевых процессоров динамично удовлетворять ресурсные требования конкретной программы. Они автоматически предоставляют программе максимум своих свободных ресурсов, которые ей необходимы.

II. Ниже будут рассмотрены известные проекты, при обсуждении которых остановимся лишь на особенностях обработки данных в процессоре и способах реализации сетевого принципа. Вне нашего внимания останется ряд вопросов по осуществлению операций ввода-вывода, инструкционному репертуару, организационной деятельности и пр. С одной стороны это делается для того, чтобы сфокусировать внимание на одной конкретной проблеме, а с другой стороны причиной этого является то, что развитие сетевых процессоров находится на своем начальном этапе.

1. Архитектурная модель Денниса / Dennis /.

В [8] предлагается вычислительная структура, которая состоит из следующих основных блоков /рис.1.а/: память, операционные устройства /ОУ/, арбитражная /АС/ и распределяющая /РС/ схемы. Машинный код программы находится в памяти, которая составлена из инструкционных ячеек, состоящих /рис.1.б/ из трех регистров. Первый регистр содержит инструкцию, а второй и третий являются запоминающей средой для операндов. Когда в них поступят необходимые промежуточные результаты, ячейка возбуждается и сигнализирует арбитражной схеме,

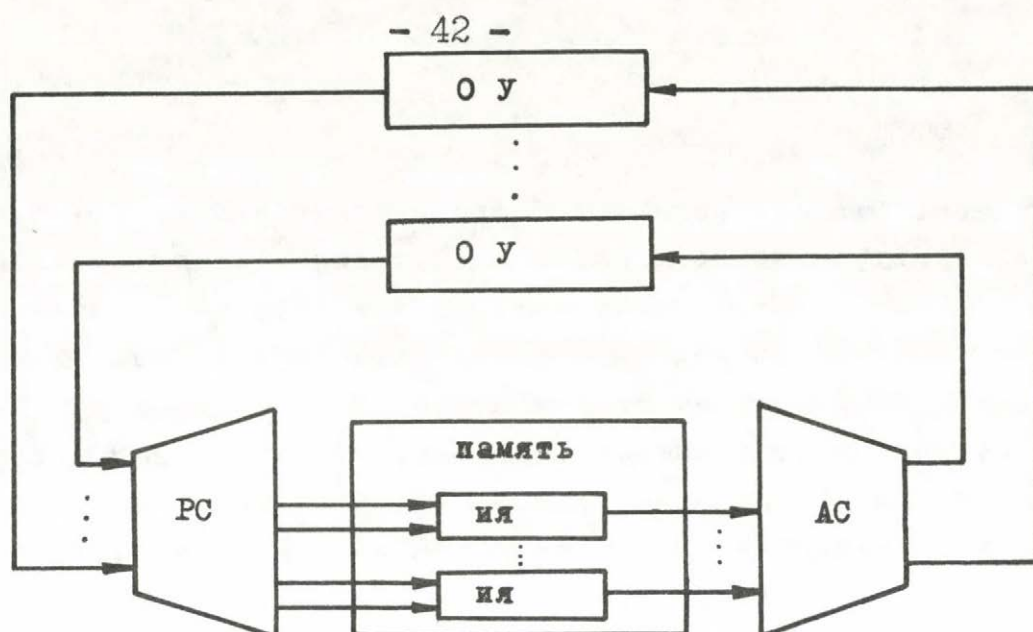


Рис.1.а. Общая схема процессора Денниса

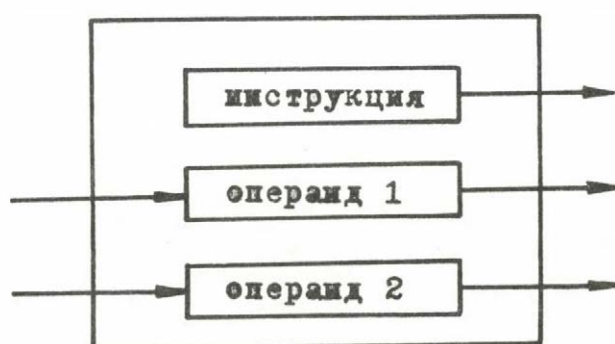


Рис.1.б. Инструкционная ячейка

что она готова послать свое содержание для обработки в виде инструкционного пакета. Арбитрирующая схема пропускает инструкционный пакет к соответствующему операционному устройству, определяемому кодом инструкции. Результат выполнения инструкции, комплектованный как информационный пакет содержит вычисленное значение и адреса регистров, к которым надо его переслать. Распределяющая схема принимает этот пакет и загружает результат вычислений в регистры соответствующих инструкционных ячеек. Когда число приемников данной инструкции больше двух применяются специальные служебные команды-расширители. Они "размножают" результат и посылают его приемникам.

Для организации ветвлений и циклов в проекте Денинса предлагаются соответствующие управляющие инструкции - "Ключ" / gate / и "Слияние" / merge /. В зависимости от состояния первого операнда, который является булевой переменной, "Ключ" пропускает или подавляет свой второй операнд. "Слияние" обеспечивает использование результатов разных инструкций в качестве операндов единственной инструкции-приемника. Чтобы структура процессора интерпретировала корректно программу, Misunas [11] предлагает предотвращать тупиковые ситуации во время выполнения программы.

2. Архитектурная модель Румбауха /Rumbaugh /. Для реализации сетевого принципа в [13] используется классическая память с адресным доступом /рис.2.а/. Функциональное разделение памяти на автономные части - память программ /ПП/, память данных /ПД/ и управляющая память /УП/ - компенсирует недостаток последовательного способа доступа к информации. Память программ сохраняет инструкции программы, в ПД находятся промежуточные результаты и вводимые данные программы а УП содержит счетчик для каждой инструкции, указывающий число ожидаемых операндов. Формат инструкции /рис.2.б/ содержит кроме кода операции /КОП/, четыре адреса: два для операндов /ОП1 и ОП2/, один адрес результата /РЕЗ/ и адрес инструкции-приемника. Адреса выполнимых инструкций сохраняются в буфере. После дешифрации в ДШ и выборки нужных операндов из ПД, инструкция выполняется в соответствующем функциональном элементе обрабатывающего устройства /ОУ/. При получении результата устройство управления /УУ/ уменьшает счетчик, соответствующий инструкции-приемника на единицу и записывает результат в ПД. Когда содержимое данного счетчика станет равным нулю, адрес соответствующей инструкции-приемника загружается в буфер для обработки, а ее счетчик устанавливается в свое первоначальное состояние. Если данный промежуточный результат должен быть передан к

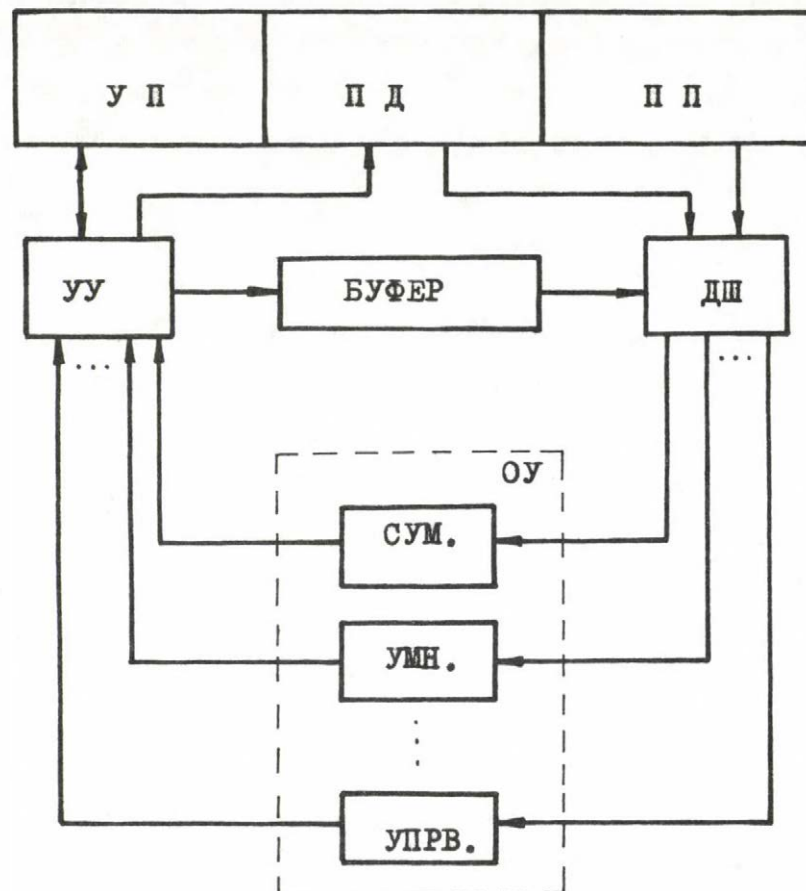


Рис.2.а. Общая схема процессора Румбауха

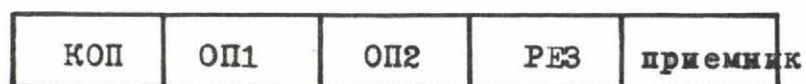


Рис.2.б. Формат инструкции

более чем одному приемнику, применяются инструкции-расширители, как и в проекте Денниса.

Румбаух предлагает два типа управляющих инструкций - "Переключатель" / Switch / и "Объединение" / Union /. "Переключатель" предназначен для реализации входа в разветвление и выхода из цикла программы, а "Объединение" - для входа в цикл и выхода из разветвления.

Для корректного выполнения программы предлагается синхронизировать промежуточные результаты на входе и выходе разветвлений и циклов.

3. Система LAU/Langage a Assignment Unique/.
Архитектурная модель системы LAU[7] в структурном отношении /рис.3.а/. состоит из локальной памяти /ЛП/, операционного устройства /ОУ/, устройства управления /УУ/ и управляющей памяти /УП/. В состав ОУ входят два основных блока - для операций обработки /ОО/ и для операций управления /ОУ/. Формат инструкции /рис.3.б/ состоит из кода операции /КОП/, адреса результата /РЕЗ/ и адресов первого /ОП1/ и второго /ОП2/ операнда. Формат данных /рис.3.в/ содержит данные /Д/

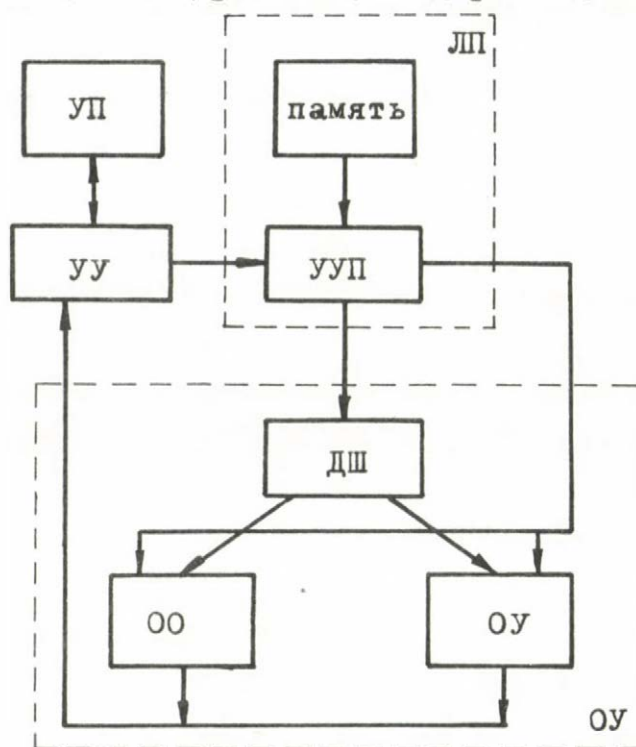


Рис.3.а. Структурная схема системы

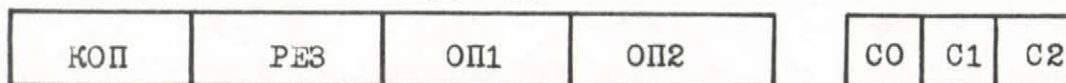


Рис.3.б. Формат инструкции

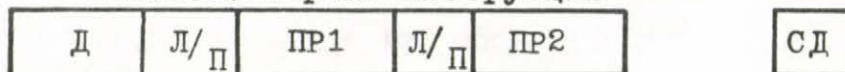


Рис.3.в. Формат данных

и адреса инструкций-приемником /ПР1 и ПР2/. Флаги Л/П указывают на то, что данный промежуточный результат используется как первый или второй операнд. Флаги СО, С1, С2 и СД применяются для реализации сетевого принципа и для организации ветвлений и циклов в программе. Они находятся в управляющей памяти. Флаги С1 и С2 указывают на наличие необходимых для каждой инструкции операндов. Флаг СО запрещает выполнение лишней ветви программы и повторное включение уже выполненных инструкций. Обработка инструкции происходит тогда, когда каждый из ее флагов находится в единице. При этом СО сбрасывается. Результат загружается в локальную память, устанавливая в единицу соответствующие флаги С1 и С2 двух приемников, а также флаг СД. Последний обеспечивает реализацию принципа однократного присваивания - только одна инструкция в программе имеет возможность присвоить значение каждой переменной. Специальная инструкция служит для увеличения числа приемников данного промежуточного результата.

В проекте системы LAU предусмотрены управляющие инструкции ACT, LOOP и CASE. Первая из них восстанавливает определенную ветвь программы для повторного выполнения /сбрасывает СД и устанавливает СО/. Цикл строится при помощи LOOP - инструкции, а ветвление - посредством CASE- инструкции. В этом случае необходима синхронизация промежуточных результатов на входе и выходе разветвления и цикла.

4. Архитектурная модель с ассоциативной памятью. Для реализации сетевого принципа в [4] используется ассоциативная память. Общая схема структуры процессора, который реализует такую модель управления, показана на рис.4.а. Процессор включает обрабатывающее устройство /ОУ/, память программ /ПП/ с ассоциативным доступом для сохранения сетевой программы, память данных /ПД/ с адресным доступом, в которой записываются вводимые данные, промежуточные и

конечные результаты, устройство управления /УУ/ и собира-
тельный /СБ/ и распределительный /РБ/ буферы. На рис.4.б.
показан формат инструкции, который содержит поля для номе-
ров первого /Н1/ и второго /Н2/ операнда и для кода опера-
ции /КОП/. Формат данных состоит из полей для операнда /Д/,
кода условия /КУ/ и счетчика актуальности /СА/, указывающий
число невыполненных инструкций, использующих этот операнд.

Для организации ветвлений и циклов в рассмат-
риваемой архитектурной модели [1] предлагается применение
аппарата обратных связей. В ПП имеются три регистра совпа-

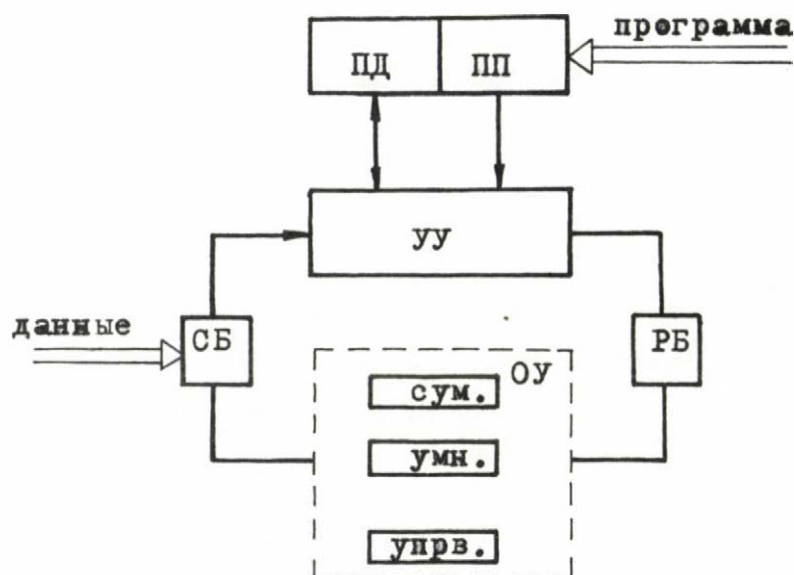


Рис.4.а. Общая схема процессора с
ассоциативной памятью

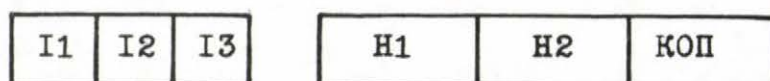


Рис.4.б. Формат инструкции

Д	КУ	СА
---	----	----

Рис.4.в. Формат данных

дения - I1, I2 и I3, каждый разряд которых соответствует одной ячейке памяти. I1 и I2 содержат флаги первого и второго операнда, а I3 предназначен для обеспечения обратных связей.

В начальном состоянии сетевой программы для всех инструкций $I1=I2=0$, а $I3=1$. I1 и I2 устанавливаются в единицу при получении соответствующих промежуточных результатов или при вводе данных. Если для некоторой инструкции соблюдается условие выполнимости /все флаги находятся в единичном состоянии/, она посылается вместе со своими операндами через РБ для обработки в ОУ. При этом все флаги инструкции сбрасываются. После вычисления результата устанавливаются соответствующие флаги I1 и I2 всех инструкций-приемников. При каждой выборке операнда из ПД его счетчик актуальности уменьшается на единицу. Установка флага I3 каждой инструкции происходит, когда счетчик актуальности ее результата станет равным нулю.

В проекте предлагаются три типа управляющих инструкций - "Ветвление", "Цикл" и "Коллектор". Инструкции "Ветвление" и "Цикл" предназначены для переключения передачи данного промежуточного результата и различным ветвям программы в зависимости от КУ. "Коллектор" обеспечивает пересылку результата двух инструкции к единственному приемнику. При ветвлениях или циклах обратные связи охватывают их целиком, а не только одну инструкцию. Таким образом структура обеспечивает корректную интерпретацию программы.

III. После рассмотрения отличительных особенностей известных проектов сетевых процессоров, остановимся на вопросе, в какой степени каждый из них выполняет сформулированные основные требования.

Реализация сетевого принципа, которая обеспечивает выполнение требования параллелизма на уровне оператора осуществляется путем указания информационных связей между инструкциями. Для решения этой задачи можно применить два подхода. Первый из них, характерный для проектов Денниса, Румбауха и системы LAU, использует инструкции-приемники данного промежуточного результата. При наличии более, чем двух приемников надо применять инструкции-расширители, а это уменьшает степень параллелизма сетевых программ. В процессоре с ассоциативной памятью информационные связи указываются посредством инструкций, предшествующих данной инструкции. Это дает возможность преодолеть вышеупомянутый недостаток первого подхода.

Параллелизм, который получается во время выполнения, в большой степени зависит от воспринятой организации памяти. В архитектурных моделях Румбауха и системы LAU предусмотрены памяти с адресным доступом, которые характеризуются своим последовательным действием. Чтобы преодолеть этот недостаток Деннис предлагает разбивку памяти на отдельные инструкционные ячейки, действующие независимо друг от друга. Такое решение ведет к усложнению организации и арбитражной и распределяющей схемы. Ассоциативная память, применяемая в четвертом проекте, дает возможность для параллельного доступа до выполнимых инструкций. Одновременно с тем она обеспечивает устранение недостатков остальных проектов.

При рассмотрении асинхронности вычислений можно заметить следующие особенности. Для организации ветвлений и циклов в проекте Румбауха и в системе LAU

используется синхронизация всех промежуточных результатов на входе и выходе ветвления или цикла. Это естественно уменьшает асинхронность вычислений. Подобный результат получается и в архитектуре Денниса, где для корректного выполнения необходимо вложить дополнительные инструкции в программу. Использование обратных связей в процессоре с ассоциативной памятью дает возможность сохранить асинхронность программы и предотвратить замедление вычислительного процесса.

Несмотря на различные организации вычислений, все рассмотренные архитектурные модели выполняют требования модульности, адаптивности и стандартизации межресурсных связей.

IV. На основе сравнения рассмотренных проектов можно сделать основной вывод, что архитектурная модель с ассоциативной памятью является самой перспективной для дальнейшего развития. Конечно, существует ряд проблем, которые ждут решения. Отметим более значительные из них. Вопрос об организации ветвлений и циклов пока еще не решен окончательно. Необходимо также углубленное исследование основных характеристик сетевых процессоров при помощи имитационного моделирования и аналитическими методами. Практического решения ждет преобразование последовательных программ в параллельные. При этом особое внимание заслуживает исследование корректности параллельных программ и сетевых вычислительных структур.

Л И Т Е Р А Т У Р А

1. В.Гетов, Т.Къичев. Организация ветвлений и циклов в сетевой вычислительной структуре с ассоциативной памятью. Труды Восьмой весенней конференции СБМ, 1979 .

2. С.Каблешков. Архитектура ЭВМ. Изд-во "Техника", София, 1978
3. В.Котов, А.Нариньяни. Асинхронные вычислительные процессы над памятью. "Кибернетика", № 3, 1966.
4. Т.Кънчев, К.Боянов. Вычислительная структура с граф-программным управлением. "УСМ", № 6, 1978.
5. D.Adams. A Model for Parallel Computations. In: Parallel Processor Systems, Technologies, and Applications. New York, 1970.
6. G.Barnes, et al. The ILLIAC-IV Computer. IEEE Trans. on Computers, C-17, N° 8, 1968.
7. D.Comte, et al. TEAU 9/7: Système LAU. Summary in English. CERT, 1976.
8. J.Dennis, D.Misunas. A Preliminary Architecture for a Basic Data-Flow Processor. Proceedings of the Second Annual Symposium on Computer Architecture. IEEE, New York, 1975.
9. J.Dennis. Packet Communication Architecture. Proceedings of the 1975 Sagamore Computer Conference on Parallel Processing. IEEE, New York, 1975.
10. R.Keller. Look - Ahead Processors. Computing Surveys, vol.7, N° 4, 1975.
11. D.Misunas. Deadlock Avoidance in a Data-Flow Architecture. Proceedings of the Milwaukee Symposium on Automatic Computation and Control, April 1975.
12. C.Ramamoorthy, H.Li. Pipeline Architecture. Computing Surveys, vol. 9, N° 1, 1977.
13. J.Rumbaugh. A Data-flow Multiprocessor. IEEE Trans. on Computers, C-26, N° 2, 1977.

СРЕДСТВА ФОРМАТИРОВАНИЯ

Й.Д.Денев, Е.К.Живкова, Р.П.Лесева, ЕЦММ, София, Болгария

Вывод информации, находящейся в ЭВМ, в виде документов, удобных для чтения, тяжелый и трудоемкий процесс даже и для программистов, пользующихся языками высокого уровня как ПЛ/1, Фортран, Алгол и т.д. При эксплуатации СУБД в БДКП необходимо создание программного обеспечения, доступного и для непрограммистов, при помощи которого можно описать форматирование большого класса достаточно сложных документов.

В Математическом институте с ВЦ, БАН, в секторе ОКТУ созданы средства для форматирования выходных документов, работающие в рамках системы БИСЕС.

Язык, при помощи которого пользуются эти средства, непроедурный, доступный и удобный для массового пользователя БДКП и освобождает его от необходимости в детальном знакомстве со структурой информации.

Его характерная особенность — широкое применение принципа умолчания, на основании использования схемных свойств базы данных. На том самом основании включены и богатые средства контроля.

Предлагаются богатые возможности редактирования и позиционирования на уровне элемента. Обеспечен автоматический выбор способа редактирования блока выходного документа и вывод заголовка, промежуточных заголовков и заголовка страницы.

Предоставлено множество средств для работы со синонимами.

Средства форматирования предоставляют пользователю две возможности: во первых — формат определяется автоматически, а пользователь указывает только то, что должно входить в выходной документ. Во втором случае пользователю предоставлено весьма полное управление форматом. Предмет настоящей работы — средства, реализующие второй способ форматирования. Определение выхода происходит при помощи языка заказов, ко-

торы является непроцедурным и позиционным. Все операторы заказа, кроме первого состоят из четырех полей фиксированной длины. Первый оператор определяет базу данных, подлежащую выводу и режимы работы при чем можно выбирать либо последовательный либо директный доступ к данным.

Заказ можно вводить с системного вводного устройства, с терминала или читать из архива системы. Все эти три способа можно совмещать в рамках одного заказа.

Выход получается на АШПУ или на терминал. За одно выполнение заказа можно получить по усмотрению пользователя до 9 экземпляров выхода, при чем возможно дублирование документов на широте страницы.

Редактирование величины на уровне элемента выполняется независимо от ее положения в строчке, на странице или в блоке выходного документа. Если пользователь не указал шаблонов реактирования, то они по умолчанию генерируются на основании схемных свойств базы данных.

Существуют несколько способов редактирования числового элемента, а редактирование символьного элемента позволяет вывести часть его и подавлять интервалов в конце символного элемента. На любом месте можно вставлять произвольный текст, задаваемый в заказе как литерал. Допустимо применение повтора.

Редактирование выходной информации состоит в размещении величин уровня элемента в строчке, задании блоков выходного документа и управлении широтой и длиной страницы. Размещение элементов в блоке осуществляется при помощи последовательного перехода от одной величины к другой, размещая каждую величину относительно предшествующей. Есть возможность задания перехода к некоторой строчке после данной или перехода к определенной строчке новой страницы. Возможно и прямо указать место величины в строчке.

В один исходный документ можно включить до 13 разных описаний

блоков. Способ редактирования определяется автоматически путем проверки логических условий, заданных пользователем.

Предусмотрены возможности выдачи заголовного текста, который располагается в начале каждой страницы и промежуточных заголовков. Описание этих текстов не отличается от описании блока выходного документа.

В рамках всего выхода или только групп блоков возможно сделать нумерацию произвольным шагом.

Есть средства определения и пользования таблиц. Существует два типа таблиц - А и В. Таблица типа А представляет набор символьных строк, а типа В - набор упорядоченных пар строк.

Сущность использования таблиц в замене значения элемента /или литерала/ на некоторую строку из таблицы. Если элемент /литерал/ имеет значение i , то он заменяется i -той строкой таблицы А. В случае таблицы типа В производится последовательное сравнение значения элемента /литерала/ с первыми частями упорядоченных пар. При совпадении значения элемент /литерал/ заменяется второй частью пары. В одном заказе можно определить до 16 таблиц.

Интенсивная эксплуатация средств системы БИС-ЕС подтвердила легкую обучаемость и их удобства.

Будущее развитие этих средств состоит в их улучшении для работы в интерактивном режиме, что является только технической трудностью.

Модификация рассмотренного набора средств дает возможность конвертирования стандартной БИС-ЕС базы в стандартный ДОС последовательный файл с редактированием.

Л И Т Е Р А Т У Р А

1. Информационные системы общего назначения. Перевод с англ. под ред. Е. Л. Ющенко. Москва, "Статистика", 1975
2. Мидоу Ч., Анализ информационных систем. Перевод с англ.

под. ред. В. М. Савинкова. Москва, "Прогрес", 1977

3. Мартин Дж., Организация баз данных в вычислительных системах. Перевод с англ. под ред. А. Л. Щерса. Москва, "Мир", 1978
4. Денев И., Лесева Р., Живкова Е., Подсистема вывода системы БИС-ЕС. Сб. докладов конференции по системам информационного обслуживания коллективов профессионально-связанных потребителей, Варна, 1977

СИСТЕМА ПЛАНИРОВАНИЯ И УЧЕТА ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ

Димитр М. Добрев, Румяна К. Киркова, Петр А. Парванов

Институт математики с ЦД, Болгарской АН

Вопросы планирования и учета вычислительных ресурсов имеют особое значение для повышения эффективности эксплуатации вычислительных центров и вычислительных комплексов. Решение этой проблемы можно достичь путем создания и поддержки соответствующую базу данных и применяя подходящую систему управления базами данных (СУБД). База данных должна состоять из записей, содержащих информацию о каждом пользовательском задании.

Разделение всей совокупности выполняемых работ, требующих применение вычислительной техники, на отдельных заданиях зависит от степени подробности, с которой администрация вычислительных ресурсов считает целесообразным ввести учет. Так как система учета сама потребляет некоторые вычислительные ресурсы, глубина указанного разделения связана затратами по эксплуатации системы.

Информация, содержащаяся в каждой записи, можно разделить на нескольких частях:

- информация о планированных ресурсах данного задания на периоде актуализации этих данных;
- информация о затраченных ресурсах данного задания в периоде актуализации этих данных;
- суммарная информация для данного задания.

Характерной особенностью двух первых частей является их динамичность - планирование ресурсов динамически меняется а информация о расходуемых ресурсах меняется при каждом выполнении данного задания. Третья часть информации является

архивом, где в конденсированном виде сохраняются суммарные данные о каждом задании во время его актуальности. Одной из самых трудоемких операций при поддержке базу данных является ее своевременное пополнение актуальными данными. При рассматриваемой ситуации значительной частью этих данных являются данные о расходуемых вычислительных ресурсах. Это обстоятельство дает возможность резко увеличить автоматизации при процессах актуализации базы данных за счет создания программных средств автоматического слежения выполнения заданий и пополнения базу данных требуемой информацией.

Правила учета расходуемых вычислительных ресурсов зависят от нормативных документов, валидных для каждой организации, как и от некоторых специфичных условий, принятых в соответствующем вычислительном центре. Это означает, что если система управления вычислительными ресурсами создается не только для одного отдельного пользователя (вычислительного центра), а с расчетом ее широкое внедрение, необходимо чтобы эта система была открытой для введения требуемых изменений. Имея ввиду, однако, что такая система связана с учетом очень больших ресурсов, необходимо предусмотреть средства защиты системы от несанкционированного вмешательства. Система управления базами данных, на основе которой создается система планирования и учета, должна обеспечить возможность получения выходную информацию разного типа. Такой выходной информацией являются ведомости о расходуемых ресурсах за данным периоде времени, как и разные справочные материалы, которые можно получить по требованию для целей управления. Очень важной частью системы, однако, является выдача учетной информации при выполнении каждого запроса каждого задания. Это дает возможность пользователю следить за ходом потребления ресурсов и рационально планировать остаток наличных ресурсов. Такая информация должна содержать:

- шифр задания;
- время (начало и конец) выполнения задания;

- плановые ресурсы данного задания на текущий период времени;
- расходуемые при этом выполнении ресурсы (процессорное время, количество обращений к разным типам внешних устройств, количество бумаги; либо обобщенные машинные единицы в зависимости от принятой в данном вычислительном центре системой планирования и учета);
- оставшие до конца текущего периода ресурсы.

Кроме того к этой информации удобно включить еще порядковый номер выполнения и конечную дату актуальности данного задания. К этой информации можно включить и некоторые сообщения, относящиеся к всем пользователям вычислительного центра, либо к некоторой их подсовкупности.

Здесь было упомянуто, что для каждого задания, для которого планируются вычислительные ресурсы, надо составить соответствующий шифр. С одной стороны этот шифр является ключом для доступа в системе, а с другой — он должен состоять из указателей, дающих возможность группирования заданий на нескольких уровнях и по нескольким критериям в зависимости от потребности администрации.

Способы задания этого шифра могут быть разные. Так например в пакетном режиме обработки задания шифр можно указать путем снабжения пакета задания специальной подпечатанной или подписанной картой, дающей право на выполнения задания. В режиме времениделения путем использования терминального доступа этот шифр является частью пароля, дающей право на доступ до вычислительных ресурсов.

При каждом выполнении система планирования и учета проверяет право на доступ и наличие требуемых ресурсов. Если задание не имеет права доступа, либо требуемые на данном выполнении ресурсы больше чем наличными, задание не выполняется и система выдает соответствующие сообщения. В противном случае система выполняет задание в зависимости от его приоритета, динамически учитывает расходуемые ресурсы и следит

за соблюдением правил эксплуатации. При нарушении этих правил (использование больше чем указанными ресурсами, несанкционированное обращение к библиотекам, файлам, базам данных и т.д.) система прекращает выполнение задания и выдает сообщение потребителю и сообщение администратору системы.

Система планирования и учета вычислительных ресурсов должна содержать простые средства для:

- открытия новых заданий;
- закрытия старых заданий;
- изменения срок актуальности;
- изменения план ресурсов;
- подготовки сообщения потребителям;

и т.д. Эти средства должны быть снабжены с определенными правилами секретности, что-бы не дать возможность потребителям вычислительных ресурсов сами менять свои плановые ресурсы.

С другой стороны такая система должна иметь возможность выдачи рядь выходных материалов (оформленные по соответствующим принятым стандартам), содержащих в различных разрезах суммарную информацию по периодам, направлениям, звенам как и детальную информацию о каждом задании. Кроме того необходимо предусмотреть простые средства создания новых выходных форм в зависимости от дополнительно возникающих потребностей каждого отдельного вычислительного центра.

В Институте математики с Вычислительным центром Болгарской Академии Наук с 1974 года проводятся разработки и эксплуатацию систем планирования и учета вычислительных ресурсов. Одна такая система, с названием АСОБЗ2, была создана в 1975 году для ЭВМ МИНСК-32. С 1976 года такая система, с названием АСОБЕС, была разработана и для ЕС ЭВМ. С течением времени эти системы, и в особенности АСОБЕС, были усовершенствованы - их возможности и степень автоматизации их эксплуатации были развиты до описанных в предыдущих абзацах.

Система АСОБЕС использует стандартные возможности операционных систем ДОС/ЕС 1 и ОС/ЕС 2,3, а для создания и поддержки базу данных - СУБД БИСЕС 4.

В настоящее время система АСОБЕС эксплуатируется как в операционной системе ДОС/ЕС, так и в операционной системе ОС/ЕС с одной стороны, а с другой - в пакетном режиме и в режиме времемеления в терминальном комплексе. Этот терминальный комплекс создан в Вычислительном центре института математики Болгарской АН и является первым этапом создания вычислительного центра коллективного пользования Болгарской АН (ВЦАП-БАН). В системе предусмотрены возможности ее работы в условиях многомашиной системы, как и в вычислительных сетях.

ЛИТЕРАТУРА

- 1 Девис, У. - Операционная система (функциональный подход), Москва, Мир, 1980
- 2 Медник, С., Дж. Донован - Операционные системы, Москва, Мир, 1978
- 3 Даниловичин, В.П., Б.В. Митрофанов, Б.В. Одинцов, Г.В. Пеледов - Операционная система ОС/ЕС (справочное пособие), Москва, Статистика, 1980
- 4 Добрев, Д.М., И.Д. Денев, Р.К. Киркова, А.М. Ескенази, И.Б. Швертнер, П.А. Парванов - Архитектура и основные принципы СУБД БИСЕС, Москва, сб. Проблемы информатики (в печати)

ДОСТУП К ЗАПИСЯМ В СУБД БИСЕС

Д.М. Добрев, Институт математики, София

И.В. Швертнер

В статье рассматриваются некоторые улучшения в возможностях адресации записей в СУБД БИСЕС, разработанной в Институте математики Болгарской Академии наук. В результате практических требований был создан проект дополнительных программных средств для осуществления доступа к данным в базе данных БИСЕС. Эти средства не заменяют существующие, а только предоставляют дополнительные возможности и удобства, которые имеют важное значение для некоторых информационных систем, разработанных с применением СУБД БИСЕС.

Система БИСЕС позволяет как последовательный, так и прямой доступ к записям в базе данных. Прямой доступ осуществляется заданным в системе соотношением между номером записи и его местоположением в базе данных. Прямой доступ по номеру записи, с одной стороны, имеет преимущества, свойственные для этого типа доступа. Однако с другой стороны он имеет и некоторые специфические недостатки:

- фиксированная длина номера;
- номер может быть только числом;
- создание новых записей или возможное устранение старых записей связывается с физическим местом. Это может привести к реорганизации базы данных.
- в большинстве случаев потребитель вынужден поддерживать свои, характерные для базы данных ключи и тогда доступ должен быть обеспечен по этим ключам, что не обеспечено прямым доступом.

Что бы предоставить потребителям БИСЕС средства, неимеющие вышеупомянутые неудобства, необходимо было создать средства

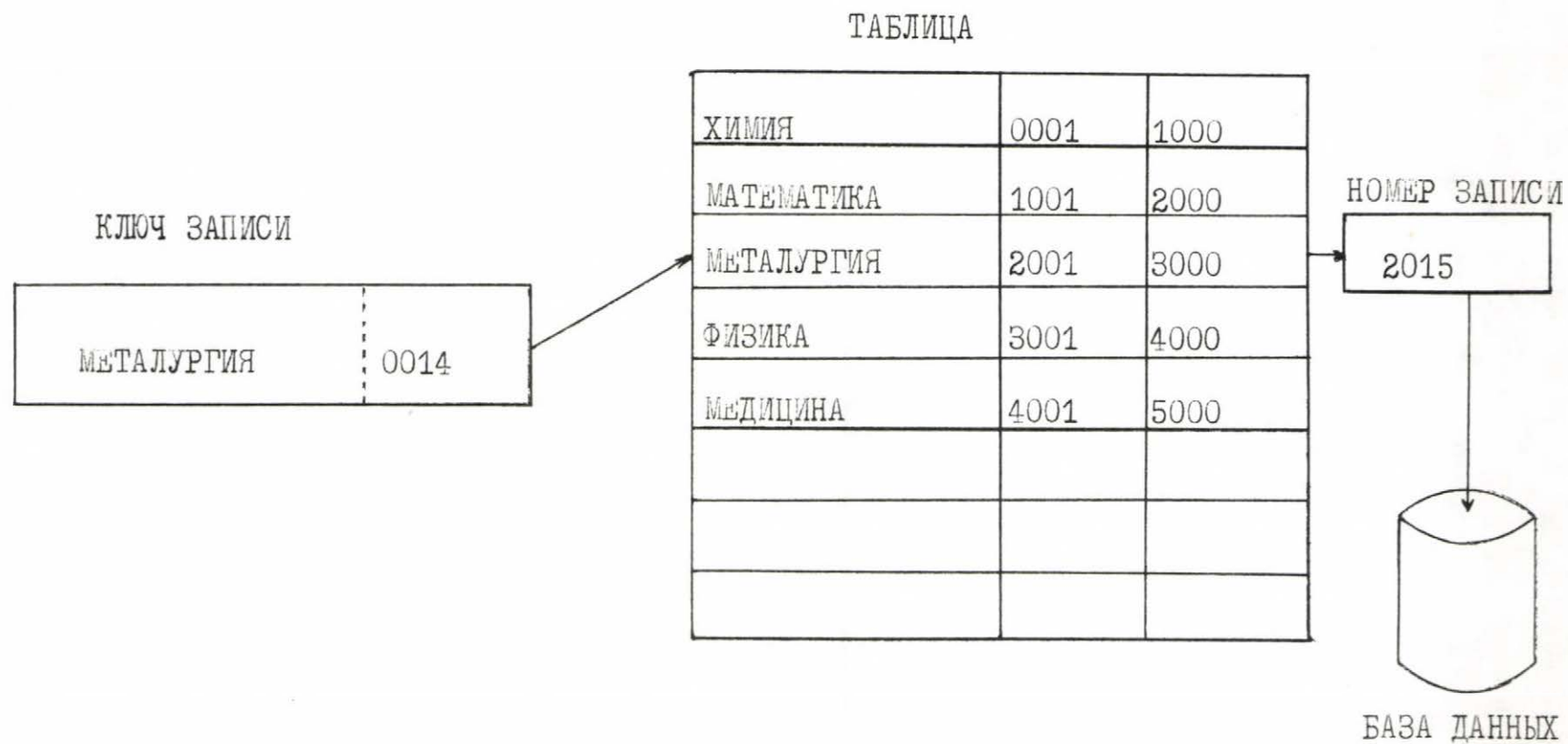
для расширения возможностей доступа к записям. При этом необходимо было обеспечить полную совместимость старых и новых средств. К функции обновление была добавлена возможность определения номера записи по таблице. Таблица создается потребителем и обеспечивает однозначное соответствие между любым допустимым ключом записи и номером записи в базе данных БИСЕС. Ключ записей является составной частью записи. Он состоит из двух частей: идентификатора и порядкового номера. Идентификатор является строкой символов любой фиксированной длиной. Он определяет соответствие между записью и некоторым интервалом номеров записей в базе данных. Интервал номеров задается своими границами, то есть самым маленьким и самым большим номером в интервале. Порядковый номер определяет отклонение номера записи от нижней границы интервала (как это показано на рисунке). В указанном примере ключу с идентификатором МЕТАЛУРГИЯ и порядковым номером 0014 соответствует запись с номером 2015.

Для автоматизации процесса нахождения номера записи по таблице, к функции обновление добавлены программные средства, предоставляющие возможность определить таблицу и записать ее в систему. Кроме того существуют возможности, которые позволяют потребителям перевключить режим работы функции обновление и заставить ее работать с использованием одного или другого метода доступа к записям. Потребителям предоставлены средства, помогающие ему как при первоначальном создании таблицы, так и при возможных ее изменениях. В случаях, когда это необходимо, можно применить созданные средства реорганизации базы данных в соответствии с новой таблицей.

Разработаны также средства, которые по созданной базе данных строят таблицу и совершают реорганизацию базы данных в соответствии с таблицей, обеспечивая доступ по ключу к записям.

Необходимо отметить, что созданное средство доступа позволяет и введение синонимов, т.е. доступ к одной и той же записи или группе записей через разные ключи.

Возможно применять и другие методы доступа к записям в базе данных БИСЕС. Общее для них является обстоятельство, что



ключ состоит из двух частей. Первая часть - это идентификатор который является строкой знаков с любой зафиксированной длиной. Идентификатор определяет интервал номеров записей, которые соответствуют идентификатору. Вторая часть ключа может быть использована для нахождения номера записи в интервале методом рендомизации. При этом возникает проблема определения подходящей функции рандомизации, а также вопрос о синонимах. Другая возможность, это использовать порядковый номер для создания записи на первом свободном месте. При обработке записей, созданных таким образом, будет использован полный перебор записей в интервале, определенном идентификатором.

Созданные средства доступа успешно применены в нескольких информационных системах, созданных с помощью СУБД БИСЕС. Опыт их эксплуатации показывает, что разработанные средства значительно облекчают потребителей при организации и доступе к использованным ими базам данных.

ЯЗЫК МАНИПУЛИРОВАНИЯ ДАННЫМИ СИСТЕМЫ УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ LINDA

В. Жечковски

Институт основ вычислительной техники ПАН, Варшава, ПНР

I. ОБЩАЯ ХАРАКТЕРИСТИКА СИСТЕМЫ LINDA

В течение последних трех лет в Институте основ вычислительной техники ПАН была создана система управления базой данных СУБД LINDA. Она реализована на ЭВМ ODRA серии 1300, компатибельных с IC1 1900. Система LINDA является системой однозадачной, централизованной, замкнутой (self contained) с одним уровнем независимости данных и программ - уровнем схемы. Главную роль в концепции этой системы играет язык описания данных, основанный на лингвистическом подходе к теории баз данных (Linguistic Data Approach) [1].

Система оснащена следующими главными модулями:

- управляющим,
- компилятором языка описания данных,
- ввода данных,
- обновления базы данных,
- компилятором языка манипуляции данными,
- модулями для реализации специальных функций, таких как сортировка, поиск, объединение файлов, печать данных.

Основные функции управляющего модуля это:

- управление ходом обработки данных,
- открытие/закрытие файлов,
- управление печатью,
- задание параметров другим модулям и их запуск.

В системе LINDA существует несколько языков, предназначенных для разных целей, среди них имеются языки описания, идентификации и манипулирования данными. Язык манипулирования данными СУБД LINDA тесно связан с языками описания и идентификации данных.

Язык описания данных СУБД LINDA разрешает:

- описать практически любое количество типов записей,
- определить повторяющиеся группы,
- декларировать данные, которые могут не выступить,
- декларировать небольшие числовые пределы,
- декларировать преобразование чисел и дат на бинарную форму во время ввода данных.

Язык описания данных в его настоящем варианте не дает возможности определять связей между записями.

Основные синтаксическо-семантические конструкции языка идентификации это идентификаторы, формулы и термы, которые создают непроцедуральный язык. Выражения языка идентификации позволяют выделять данные из базы данных. Определение фактического значения идентификаторов, формул и термов происходит во время реализации конструкции, в которых они выступают, например, инструкций языка манипулирования данными, выражений управляющего модуля и других. В этом процессе обычно участвует только часть базы данных, именно эта часть базы данных, которая находится в операционной памяти вычислительной машины. Называем ее полем зрения. Содержимое поля зрения можно изменять при помощи управляющего модуля или при помощи языка манипулирования данными.

Вышеприведенные главные конструкции языка идентификации принимают следующие значения:

- идентификатор: список адресов данных, принадлежащих к полю зрения, например, значение идентификатора [РАБОТНИК: [ФАМИЛИЯ] = ИВАНОВ] это список адресов записей работников с фамилией Иванов.
- терм: список констант или элементарных значений, определенных на основании поля зрения, например, терм I, [РАБОТНИК: [ФАМИЛИЯ] = ИВАНОВ; ИМЯ] принимает значение списка, в состав которого входят: константа I и все имена работников с фамилией Иванов. Значение термина [РАБОТНИК: [ФАМИЛИЯ] = ИВАНОВ] это пустой список значений. Именно этот терм определяет все данные о названии РАБОТНИК, но они неэлементарные, так как в их состав входят другие данные, например, данная

ФАМИЛИЯ.

Для некоторых целей в СУБД LINDA употребляется расширенное определение терма, в котором допускаются неэлементарные значения терма. В таком случае значением вышеприведенного терма стали бы все структуры, подчиненные данным РАБОТНИК, в которых элементарная данная ФАМИЛИЯ содержит значение ИВАНОВ.

- формула: одно значение из множества {истина, ложь}.

Например, формулой является выражение

[РАБОТНИК; ФАМИЛИЯ] = ИВАНОВ

которое принимает значение истина, если актуально в поле зрения находится хотя одна запись работника с фамилией Иванов и значение ложь в другом случае.

Заметим, что каждый идентификатор можно считать термом, которого значением является множество содержащихся находящихся под адресами, показанными идентификаторами.

Процессы реализации многих инструкций управляющего модуля как тоже компиляция и выполнение манипуляционных программ требуют динамического распределения оперативной памяти машины. В СУБД LINDA эту задачу выполняет система динамической памяти, описанная в работе [2]. Свойства системы LINDA описаны более подробно в [3], [4], [5].

2. ОБЩАЯ ХАРАКТЕРИСТИКА ЯЗЫКА МАНИПУЛИРОВАНИЯ ДАННЫМИ

Язык манипулирования данными СУБД LINDA является языком программирования высшего уровня. Основные синтаксическо-семантические единицы языка это инструкции, последовательность которых составляет манипуляционную программу. Манипуляционные программы предназначены для обработки физических записей о структуре, присущей системе LINDA.

Язык манипулирования запроектован, главным образом, как средство для организации вывода в требуемом графическом виде, определенных отчетов, касающихся входных массивов данных. Следующие группы инструкций и синтаксическо-семантических конструкций языка обеспечивают реализацию вышеупомянутой главной функции языка:

- обращения к базе данных,
- инструкции вывода результатов обработки данных,
- инструкции чтения данных из перфокарт, которые примерно можно использовать для управления выполнением манипуляционных программ,
- инструкции создания данных, которые в программе могут например, играть роль переменных потребителя,
- инструкции присваивания,
- условные инструкции и инструкции прыжков,
- арифметические выражения в обобщенном виде - термы,
- булевы выражения в обобщенном виде - формулы,
- выражения, идентифицирующие данные - идентификаторы,
- инструкции, выполняющие специальные функции, примерно обращения к управляющему модулю, статистические инструкции [4] и другие,
- структурные инструкции типа инструкций, выступающих в языке PASCAL (case, while, repeat, for, процедуры) [5].

Язык манипулирования данными позволяет динамично определять устройство для вывода результатов действия программ. Эти результаты могут быть непосредственно печатаны на бумаге или могут попасть в формате печатающего устройства на магнитную ленту. Файл, который в этом случае возникнет на магнитной ленте называем файлом печати. В системе существует спе-

циальная инструкция управляющего модуля, которая позволяет печатать данные, находящиеся в таком файле в соответствующей, заданной программной форме.

Инструкции языка манипулирования данными делятся на простые и сложные, то есть инструкции, которые имеют тело, содержащее любое количество инструкций языка манипуляции. Каждая инструкция манипуляционной программы может быть обозначена меткой, которую можно использовать в инструкциях прыжков к обозначенной ее инструкции.

Грамматический разбор манипуляционной программы совершается методом из верха вниз (top down). Для этой цели употребляется соответствующий состав рекурсивных процедур.

3. ОБЩИЕ СВОЙСТВА МАНИПУЛЯЦИОННОЙ ПРОГРАММЫ

Манипуляционная программа является в основном последовательностью инструкций языка манипулирования данными; программа не имеет блочной структуры. В программе не выступают никакие статические декларации структур данных. Структуры данных определяются динамически во время создания данных. Даже обыкновенные переменные потребителя считаются такими же, конечно, упрощенными, записями и можно их генерировать в программе аналогичным образом как другие более сложные записи. В программе допускается также динамически удалять целые или определенные части данных. Это свойство дает возможность ограничить размер операционной памяти, нужной для выполнения задачи.

Программа имеет свое собственное название, при помощи которого можно ее идентифицировать в обращениях к управляющему модулю, таких как вызвание компиляции, выполнения или удаления программы.

Манипуляционные программы реализуются в компиляционном порядке, то есть на основе входного текста в процессе компиляции генерируется программа в виде внутреннего кода вычислительной машины. Одновременно наступает зарядка этого кода в операционную память машины. С этого момента программа может

быть выполнена многократно. Ее можно выбросить из памяти, употребляя некоторые инструкции управляющего модуля в частности, вызывая компиляцию другой программы.

Система допускает компиляцию программ, записанных на перфокартах или на магнитной ленте.

Язык манипулирования реализован таким образом, что имеются средства для контроля программ на этапе компиляции, а также и выполнения. Компилятор сигнализирует синтаксические ошибки и тоже доставляет другие важные информации. Между прочим, он указывает список всех таких названий и констант, примененных в программе, которые не принадлежат к списку названий и констант, выступающих в обязывающем выражении описания данных. Это разрешает найти некоторые семантические ошибки в программе еще до ее выполнения. Допустим, например, что в списке, данном компилятором, находится название РАБОТНИК и мы знаем, что обязывающее выражение описания данных описывает записи о названии РАБОТНИК. Конечно, на основе этих информации мы можем догадываться, что в каком-то идентификаторе, выступающем в тексте манипуляционной программы, мы ошибочно обращаемся к не существующей записи РАБОТНИК вместо к записи РАБОТНИК.

Независимо от существующей возможности управления количеством информации, доставляемым компилятором, потребитель может выбрать один из двух порядков компиляции:

- порядок нормальный, в котором получается обыкновенный выходной код, применяемый в нормальной эксплуатации программы,
- порядок специальный, в котором получается обогащенный выходной код, предназначенный для этапа проверки правильного действия программы.

Потребитель может динамично (во время действия) или статично (до старта) определить состав ситуаций, которые считает для данной программы ненадежными, аварийными или вообще интересными по другим причинам, и контролировать их **выступление**.

Выполняя программу, компилированную в специальном порядке, можно добавочно получить сигнализацию перехода программы через все инструкции, обозначенные метками, сигнализацию ста-

рта выполнения сложных инструкций и процедур, а также значения булевых условий в условных инструкциях. Программа задерживается во всех моментах, когда выступит любая, заданная потребителем, подозрительная ситуация. Большинство ненормальных ситуаций, которые могут появиться во время действия манипуляционной программы, возникает вследствие многозначности термов, избытков во время арифметических операций или выступления неподходящих значений данных, что связано с отсутствием в языке формальных деклараций типов данных.

В моменте старта любой манипуляционной программы поле зрения является пустым или содержит часть базы данных, введенную в оперативную память машины инструкциями управляющего модуля. Обычно ранняя зарядка поля зрения имеет в виду установить что-то в роде базы, использованной в выражениях языка идентификации данных.

4. ОСОБЕННОСТИ УПОТРЕБЛЕНИЯ ВЫРАЖЕНИЙ ЯЗЫКА ИДЕНТИФИКАЦИИ ДАННЫХ В МАНИПУЛЯЦИОННОЙ ПРОГРАММЕ

В инструкциях языка манипуляции выступают выражения языка идентификации согласно следующим принципам:

- всегда там, где требуется показать место расположения данных, употребляются идентификаторы,
- в случаях, когда требуется определить значения данных или константы - применяются термы,
- если надо построить условное выражение - применяются формулы.

Специфика инструкций языка манипулирования данными возникает из многозначного характера термов, которые в них выступают. В зависимости от конкретного содержания поля зрения во время исчисления данного терма, может он принять одно или больше значений, но он может также не принять никакого значения.

В языке допускаются различные операции на множествах элементарных значений, заданных термами, между прочим:

- арифметические операции плюс, минус, умножение, деление,

- операции на множествах, такие как пересечение множеств, разность множеств, объединение множеств и т.д.,
- отношения между множествами - равенство, подчинение и т.д.,
- арифметические сравнения, такие как больший, меньший, равный.

На все эти операции огромное влияние оказывает вышеприведенное свойство термов - их многозначность.

Например, для арифметических операций или операций сравнения из этого свойства возникает важный вопрос: каким образом они должны выполняться? какой, например, должен быть результат арифметической суммы двух множеств значений, заданных соответствующими термами?

В актуальном варианте системы LINDA принято принцип, что в результате любой арифметической операции на двух термах возникает новый терм. Если допустим, что первый терм это множество значений T_n , а второй - это множество значений T_m , тогда можно сказать, что элементы множества значений терма - результата, это результаты исполнения данной операции (здесь плюс) на всех парах из картезианского продукта $T_n \times T_m$, для которых операция возможна (система автоматически совершает необходимые преобразования внутренних физических видов данных).

При таком подходе, к сожалению, теряется контроль над количеством значений в множестве выходного терма, потому что не всегда точно известно, сколько будет таких пар значений, для которых операция не выполнится. Это может оказаться источником некоторых трудностей в правильном определении дальнейших операций на этих выходных ^{данных}, в частности - инструкций печати и присваивания. С этой точки зрения, конечно, лучшим решением является такое, в котором каждая пара значений доставляет точно одно значение до множества результатов. В таком случае надо бы ввести понятие специального пустого значения, допустим, NIL. Оно будет результатом арифметической операции на каждой такой паре значений, для которой данная операция нормально не дается выполнить. Однако, и этот подход имеет свои недостатки: во первых, увеличивает размер опе-

рационной памяти, нужной для хранения множеств значений термов, среди которых много значений может быть NIL, во вторых, увеличивает время выполнения операций на термах, потому что добавочно выступают действия на пустых элементах NIL, в третьих, усложняет исполнительные программы.

5. ИНСТРУКЦИИ, СВЯЗАННЫЕ С ДОСТУПОМ К БАЗЕ ДАННЫХ

Основные типы операций, касающиеся записей базы данных, практически одинаковы для всех информационных систем с базой данных. Это, конечно, операции создания, хранения и поиска данных. В языке манипулирования данными СУБД LINDA эти функции выполняются ответственно следующими инструкциями: #ASSIGN и #CREATE, #STORE, #GET. Новые данные, возникающие вследствие выполнения инструкций языка #ASSIGN или #CREATE автоматически присоединяются к полю зрения.

Инструкция #ASSIGN является, прежде всего, инструкцией присваивания: вместо (элементарных или неэлементарных) значений, показанных идентификатором, подставляются данные структуры, определенные термом. Однако, если в инструкции #ASSIGN идентификатор заменим каким-то названием, инструкция станет инструкцией создания и, одновременно, присваивания данных. Именно для каждого элементарного значения терма она создает элементарную данную, которой названием является название, выступившее в инструкции, а значением — очередное значение терма. Например, инструкция

#ASSIGN ФАМИЛИЯ, [РАБОТНИК; ФАМИЛИЯ];

создает данные ФАМИЛИЯ. Значением каждой такой данной стает одно значение из множества всех фамилий, выступающих в данных РАБОТНИК. Возникнувшие вследствие вышеприведенной инструкции данные ФАМИЛИЯ можно, например, присвоить полям ФАМИЛИЯ в данных АВТОР, применяя инструкцию #ASSIGN в виде инструкции присваивания, как следует:

#ASSIGN [АВТОР; ФАМИЛИЯ] , [ФАМИЛИЯ] ;

В инструкции присваивания #ASSIGN требуется, чтобы, во первых, идентификатор показывал столько данных, сколько зна-

чений принимает терм, и, во вторых, чтобы размер операционной памяти занятой *i*-той данной, указанной идентификатором, был не меньший, чем размер *i*-того значения термина, которой в это место подставляется. Инструкция присваивает данные, пока эти условия совершены. В моменте, когда они первый раз не будут совершены, инструкция кончится и передает об этом сигнализацию согласно требованию потребителя (пункт 3). Выполнение инструкции присваивания **#ASSIGN** не изменяет размеров данных, показанных идентификатором, хотя новые значения, подставленные в их место, могут занимать меньше памяти.

Второй, гораздо более мощной инструкцией создания данных является инструкция **#CREATE**. Она создает одну данную о неограниченно сложной иерархической структуре. Эту структуру, названия и значения элементарных и неэлементарных данных, из которых вся данная состоится - описывается в инструкции. Значения элементарных и неэлементарных данных, то есть подструктур - задаются терминами. Названия данных, кроме этих, которые выступают в неэлементарных подструктурах, заданных терминами - поданы явно. Общая синтаксическая форма инструкции **#CREATE** имеет рекурсивный характер, также как создаваемая ей данная. Основной единицей этой рекурсии является описание данной, которое может выступить на любом уровне иерархии. Последовательность: двоеточие и открывающая скобка начинают определение при помощи термина простого значения или подструктуры. Открывающая скобка без знака двоеточие обозначает переход к подчиненным данным на следующем уровне иерархии. Пример:

```
#CREATE      РАБОТНИК (ФАМИЛИЯ: (ИВАНОВ)  ИМЯ: (НИКОЛАЙ)  
                СЕМЬЯ ( ЖЕНА:([ЖЕНА])  ДЕТИ: ([ДЕТИ])));
```

Вышеприведенная инструкция **#CREATE** создает данную РАБОТНИК, у которой две элементарные данные ФАМИЛИЯ и ИМЯ (содержащие значения ИВАНОВ и НИКОЛАЙ) и неэлементарная данная СЕМЬЯ. Данная СЕМЬЯ состоит из данных ЖЕНА и ДЕТИ, значениями которых являются соответственно структуры, заданные терминами [ЖЕНА] и [ДЕТИ]. Эти термы могут представлять как простые значения, как также совсем сложные неэлементарные структуры.

Для реализации операции хранения данных в языке манипуляции запроецирована инструкция **#STORE**. Ее параметром является идентификатор. Он определяет данные, которые надо хранить. Эти данные в виде физических записей передаются к актуальному выходному файлу на диске или ленте или к полю зрения, именно этой его части, которая предназначена для хранения части базы данных постоянно нужной системе. При помощи инструкции **#STORE** можно выделить из поля зрения определенные данные в целом или их части и оформить их в обыкновенную физическую запись, присущую системе LINDA.

Примеры. **#STORE [РАБОТНИК];**
#STORE [РАБОТНИК; ФАМИЛИЯ];

Вследствие выполнения первой из вышеприведенных инструкций хранятся в выходном файле все данные РАБОТНИК, а вследствие второй - только их поля ФАМИЛИЯ.

В настоящем варианте языка манипуляции предусмотрена одна инструкция поиска записей из базы данных - инструкция **#GET**. Она реализует последовательный просмотр актуальной входной базы данных, доставляя к полю зрения каждый раз одну запись. Для этой записи выполняется последовательность инструкций языка манипулирования, которая называется телом инструкции **#GET**. После выполнения тела запись удаляется из поля зрения. Несколько допускаемых синтаксических видов этой инструкции разрешает, во первых: выполнять тело инструкции **#GET** для всех записей из базы данных или только для записей определенных идентификатором, выступающим в инструкции; и во вторых: окончить поиск записей после просмотра всей базы данных или в моменте, в котором значение определенной формулы станет истиной.

I N T E P A T Y P A

1. Subieta K.: "Linguistic Approach to Database Theory: DDL's for Hierarchical Model". Information Systems Vol. 3 pp. 203 - 208, 1978
2. Rzeczkowski W.: "Pamięć dynamiczna dla emc ODRA serii 1300". Prace IPI PAN nr 315, 1976
3. Kopeć B.; Kuta M., Rzeczkowski W., Subieta K.: "System zarządzania bazą danych LINDA", Prace IPI PAN nr 375, 1979
4. Rzeczkowski W.: "Statystyczny moduł systemu LINDA". Prace IPI PAN nr 376, 1979
5. Subieta K.: "Nowe możliwości systemu LINDA". Prace IPI PAN nr 378

КОНЦЕПЦИИ И МОДЕЛЬ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ИНФОРМАЦИОННОГО ОБСЛУЖИВАНИЯ КОЛЛЕКТИВОВ

Владимир Занев

Институт математики с БЦ БАН, София, Болгария

В секторе Математического обеспечения ИИ с БЦ, БАН коллектов сотрудников под руководством ст.н.с. П.Бърнева разрабатывает абстрактную систему информационного обслуживания коллективов (АСИОК). АСИОК модель системы информационного обслуживания при помощи ЭВМ. На основе этой модели можно строит конкретные системы предназначены для потребностей определенного коллектива – производственного, научного, обслуживающего, управляющего, учебного и т.д. [1].

Коллектив – это система, которая:

- объединяет группу людей, организованные на базе профессиональных принципов, которые работают систематически сообразно планы чтобы решать одну или более взаимосвязанные задачи;
- включает в себе ресурсы: машины, материалы, транспорт, сооружения, финансы и общительные связи;
- имеет стабильную организационную и управляющую структуру;
- основные информационные деятельности хорошо определены, регламентированы и организационно обеспечены.

АСИОК состоит из двух основных частей – информационный процессор и архив. Информационный процессор АСИОК обеспечивает ее функционирование. Он состоит из управляющей системы, системы обработки и системы обмена. Архив – это интегрированная совокупность данных и программ, рассматриваемые как таблицами. Они связаны между собой и обслуживают осно или несколько приложения оптимальным способом [1]. Информационный архив рассматриваемый от позиции базы данных содержит отдельные файлы, а также одна или более иерархические базы данных [2].

1. Распределенная информационная деятельность.

Информационный процесс – это такое отражение структуры внутренней и внешней среды систем, которое используется для сохранения качественной определенности и развития систем [4].

Информационная деятельность – сочетание из четырех основных информационных процессов: сбор информации, хранение информации, обработка информации, распространение информации.

О распределенной информационной деятельности идет речь когда информационные процессы, включены в некоторой информационной деятельности совершаются на различных географических пунктах. Это означает что разные информационные процессы осуществляются на разных местах: сбор информации делается средствами, которые находятся на одной географической точке, обработка – в другом месте, хранение – на третьем и распространение – на четвертом месте. Распределенную информационную деятельность есть и когда по крайней мере один из основных информационных процессов осуществляется на разных географических пунктах. Обработка информации, ее распространение и хранение тоже могли бы отдельно географически распределены.

Распределенные информационные системы (РИС), которые проектируются и реализуются на базе сети ЭВМ представляют одну современную тенденцию определенного интереса и важности для многих потребителей. В принципе РИС сопровождаются формированием и обслуживанием распределенных баз данных. РИС могут рассматриваться конвергентной точкой сети ЭВМ и разных модели и реализации баз данных.

2. Распределенная система информационного обслуживания коллективов (РСИОК).

Все основные элементы, составляющие одного коллектива: люди, ресурсы, информационная деятельность, управляющая и организационная структура, могли быть географически распределены.

Географическое распределение СИОК надо рассматривать в одном из следующих смыслов:

- централизованный архив и географически распределенные части и функции информационного процессора;
- географически распределенный архив и централизованный информационный процессор;
- географически распределенные архив и информационный процессор;
- централизованные архив и процессоры СИОК которые находятся в разных географических точках, соединенные между собой коммуникационными средствами с целью обеспечения потребительского доступа до каждого из них.

Потребители имеют доступ до информационного процессора и до архива каналами. Канал - это логический путь по которому идет информация от потребителей и к ним. В каналах включаются одни или другие физические устройства - коммуникационные линии передачи, модели, адаптеры, мультиплексоры, концентраторы и коммуникационные процессоры.

Архив РСИОК может быть либо:

- подразделенный архив, либо;
- централизованный архив с или без временных архивов.

Подразделенный архив - это такой тип архива, который состоит из подразделов, но рассматривается как единым логическим объектом. Подразделы содержат таблицы ориентированы к определенному приложению и географическому месту. При некоторых таблицах данного раздела возможны семантические связи (вложение) с таблицами другого подраздела. Подразделенный архив - это архив децентрализованного типа.

Основной принцип создания архива второго типа - это повторение, копирование всех таблиц или частей централизованного архива. Централизованный архив содержит все данные и программы РСИОКа. Копия отдельных частей архива образуют временные архивы. В течении определенного периода (например днем) информационная деятельность делается главным образом времен-

ными архивами. После этого периода централизованный архив обновляется (например ночью) и создаются новые копии временных архивов. Второй тип архива - это архив централизованного типа.

Кроме информационный процессор в составе информационный сети РСИОКа может участвовать и спутниковый информационный процессор. Это информационный процессор, который выполняет определенные, но не все возможные функции и обеспечивает работу хотя-бы с одним локальным временным архивом.

В зависимости от информационной деятельности - централизованная и децентрализованная, можно рассматривать и два типа РСИОКа - вертикальная и горизонтальная системы.

Вертикальная РСИОК - это такая система, при которой между компонентами информационной сети и функциями, которые они исполняют есть иерархические связи. Информационный процессор и архив структурированы иерархически таким способом что каждая компонента системы управляется членом иерархии, который на более высоком уровне. В составе вертикальной РСИОК участвуют спутниковые информационные процессоры. Функции вертикальной РСИОК распределяются иерархично по информационной сети. При распределении частей информационного процессора и его функции должны быть выполнены следующие критерии:

- все функции должны осуществляться таким способом, что отношение цена/времени исполнения быть лучшее для этого географического пункта;

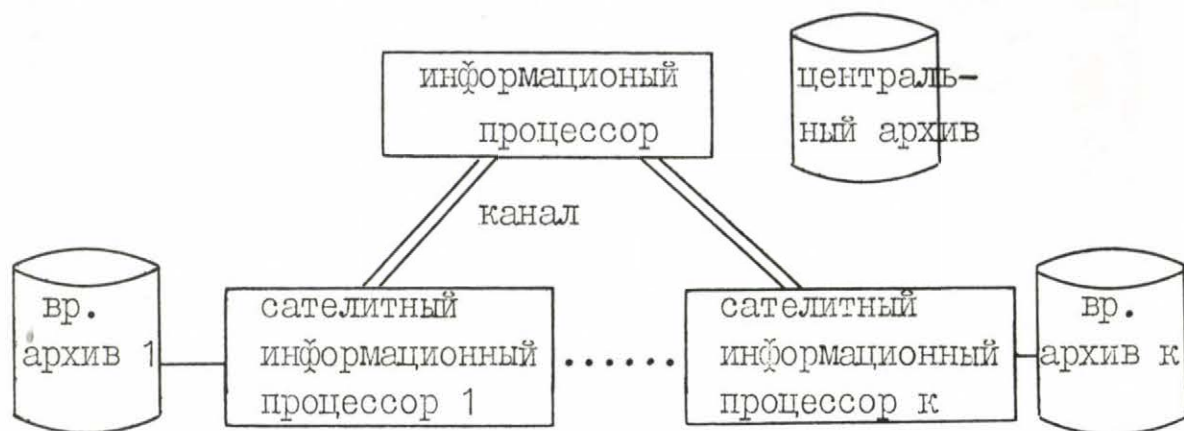
- функции, которые исполняются многократно должны быть ниже в иерархии, а те которые исполняются реже, находятся выше в иерархии;

- функции, которые требуют немного времени для ответа должны быть по возможности ниже в иерархии.

Эти критерии показывают, что система обмена РСИОК (ввод - выводные функции и функции обмена с архивом) надо включить в спутниковые информационные процессоры. С другой стороны элементы управляющей и обрабатывающей систем вертикальной РСИОК должны быть иерархически распределены между спут-

литными информационными процессорами и информационным процессором.

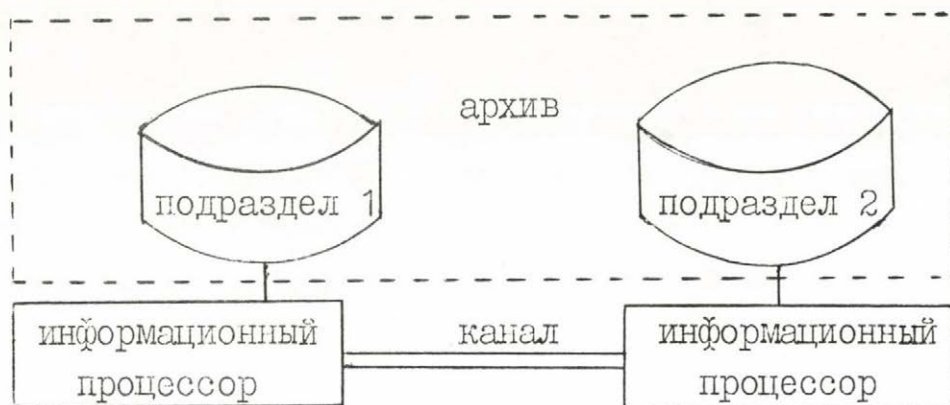
Архив в вертикальной РСИОК может быть централизованным с временными архивами или подразделенным.



Фиг. 1. Вертикальная РСИОК с централизованным архивом

Второй тип РСИОК горизонтальный. Горизонтальная РСИОК — это система, у которой все компоненты находятся на одном уровне и между ними нет иерархических связей. Информационные процессоры и каналы обеспечивают связь между двумя или более компонентами РСИОК, которые логически и функционально неразличимы. Горизонтальная РСИОК иллюстрирует одну современную тенденцию — связь между отдельными, независимыми ЭВМ с целью обеспечения общего доступа до всех информационных и аппаратных ресурсов ЭВМ. Архив горизонтальной РСИОК может быть централизованным или с подразделами.

Возможны реализации РСИОК смешанного типа. В этом случае можно рассматривать две или более вертикальные РСИОК связанные горизонтально информационными процессорами.



Фиг. 2. Горизонтальная РСМОК с подразделенным архивом

3. Протоколы в РСМОК

В РСМОК есть две группы протоколов:

- протоколы типа "потребитель - информационный процессор";
- протоколы типа "информационный процессор - информационный процессор".

Это коммуникационные протоколы на уровне передачи информации между потребителями и между потребителскими процессами. Они определяют вызов потребительских процессов, связь и взаимодействие между ними, гарантируют последовательность выполнения и прерывание связи между процессами [5].

Протоколы "потребитель - информационный процессор" управляют информационными каналами, обеспечивают и прерывают связь между потребителями и РСМОК. По идее это протоколы в одной терминальной системе. Протоколы этого типа используют таблицы описания потребителей и терминалов. Язык заявок дает средства для реализации таких протоколов. При выполнении этих протоколов обеспечиваются следующие функции: установливание связи и начала сеанса с информационным процессором, начало ввода заявок от данного потребителя для информационного процессора, конец ввода заявок, уничтожение связи и конец сеанса с информационным процессором.

Протоколы "информационный процессор - информационный процессор" управляют доступ к таблицам архива РСМОК и

совместное параллельное использование разными потребителями. Эти протоколы контролируют интерференцию потребительских заявок и наблюдают за тупиков.

Заявки использования таблиц архив распределяются следующими группами:

- заявки актуализации или управление архива: исполнение этих заявок должно быть защищено влиянием исполнением других заявок;

- заявки с защищенным доступом до таблиц архива: при исполнении этих заявок доступ до определенной таблицы блокируется для других заявок, но различные таблицы могут быть использованы одновременно такими заявками;

- заявки с незащищенным доступ до таблиц архива.

И у вертикальной и у горизонтальной РСМОК самые сложные протоколы в случае распределенного архива с копиями таблиц разных местах (во временных архивах или в подразделах). Если заявка для работы с таблицей этого архива требует обновление, протокол должен обеспечить доступ до каждого копия таблицы в распределенном архиве.

4. Модель распределенной СМОК

При помощи языка GPSS создана симуляционная модель горизонтальной РСМОК с подразделенным архивам без копии таблиц. На основе центральной обслуживающей модели одной ЭВМ разработана модель двух отдаленно связанных ЭВМ, которая обобщается легко для сети ЭВМ произвольным числом машин.

Симуляционная модель состоит из:

- модели аппаратного оборудования;
- модели математического обеспечения;
- модели задания.

Время в модели является дискретной величиной и измеряется миллисекундами.

В модели аппаратного оборудования системы участвуют 25 разных устройств, которые задаются реальными параметрами, соответствующие на устройствах конфигурации машины серии ЕС. Каждой из этих параметров можно изменить чтобы исследовать

их влияние на результаты моделирования. В модели аппаратного оборудования всем вероятностным параметрам и характеристикам физических устройств присвоиваются средние значения.

Модель математического обеспечения охватывает базисное обеспечение, специализированное обеспечение и распределенная СИОК. В модели базисного обеспечения моделирована основная управляющая программа Супервизор вместе с механизмом выбора приоритетной задачи и поддержка мультипрограммного режима работы. В специализированном обеспечении моделированы функции передачи данных между двумя отделенно связанными ЭВМ.

В модели РСИОК участвуют пять подсистем, четыре ввод, выводных очередей и распределенный архив без копии таблиц.

Модель РСИОК детерминированная. Каждая очередь характеризуется двумя значениями. L_q — среднее число объектов в очереди и W_q — среднее время ожидания. Дисциплина обслуживания FIFO, обслуживание — приоритетное с захватом обслуживания.

Модель задания вероятностной. Параметры и переменные, которые описывают одного задания даются вероятностными распределениями. Задание моделируется при помощи четырех непрерывных и одной дискретной функции распределения и 18 вероятностных параметров.

Вводный поток задания — статичный, но при разных прокруток симуляционной модели его можно менять. Рассмотрены вводные потоки Пуассона с разными средними значениями и потоки произвольного эмпирического типа. Вводный поток задания генерируется дискретными функциями распределения. Для идентификации закона распределения некоторых эмпирических случайных величин используется χ^2 — критерии.

Основные результаты, которые получаются работой симуляционной модели относятся к проектированию математического обеспечения РСИОК. Выясняются более важные программные модули, которые должны быть созданы, их функции и взаимоотношения с

операционной системой, обеспечение отдаленной связи и передачи данных в РСМОК.

Результаты охватывают 23 статистических исследования величин, 14 таблиц вероятностных распределении, 4 сравнительных фигур и 2 график. Они позволяют сделать анализ работы двух отделенно связанных ЭВМ серии ЕС, которые работают с РСМОК.

Некоторые более важные выводы, получены анализом результатов: самое использованное устройство это печатающее устройство, скорость передачи 4800 б/сек удовлетворяет - требования РСМОК, надежность порядка 10^{-5} б/сек достаточна, выполнение задания сильно зависит от характеристик вводного потока, дисциплина обслуживания и от последовательности поступления задания для обслуживания.

Использование симуляционной модели на этапе проектирования показывает ряд преимущества, которые позволяют иметь в виду разные решения. Параметры системы могут анализироваться и изменяться, с целью найти самые оптимальные значения, включая таких, которые не могут контролироваться в реальных условиях. При создании симуляционной модели был использован подход интеграции эмпирических исследований с аналитическими методами, которой оказался хорошо применимый для исследования и анализа работы вычислительных систем.

Л И Т Е Р А Т У Р А

- [1] П.Бърнев, Системи за информационно обслужване на колективи, Национална школа "Програмиране 77", Приморско, 1977.
- [2] П.Бърнев, Вл.Занев, Ат.Раденски, Бази от данни. Система за информационно обслужване на колективи, VIII пролетна конференция на СМБ, април 1979, Сл. Бряг.
- [3] П.Бърнев, Информация и управление, изд. "Народна просвета", София, 1978.
- [4] Гришкин И.И., Понятие информация, изд. "Наука", Москва, 1973.
- [5] Вл.Занев, Комуникационни протоколи, Сборник доклади 1975 - 1977, Семинар по информатика, изд. БАН, 1978, София.

ЭКВИВАЛЕНТНОСТЬ МОДЕЛЕЙ ДАННЫХ
В РАССПРЕДЕЛЕННЫХ БАЗАХ ДАННЫХ

Ф. Златарова

Институт Математики Болгарской Академии Наук
София, Болгария

Формальное описание информационных ресурсов одной распределенной базы данных (РБД) при помощи соответствующих ей информационных и функциональных графов, задающих структуру, свойства и взаимоотношения этих информационных ресурсов, является объектом настоящей работы. Исследование этих вопросов имеет смысл, если существуют необходимые аппаратные и программные возможности реализации и эксплуатации одной РБД. Такие возможности предоставляет вычислительный центр коллективного пользования.

Основное понятие для неоднородных РБД — это понятие "модели данных", которое определяется разными способами [2,3,4]. Модель данных вполне определена и зависима от соответствующих ей языка определения данных и языка манипулирования с данными. Если в РБД представлена только одна модель данных, РБД называется однородной, в противном случае — неоднородной.

Независимо от содержащихся в одной РБД моделей данных вводится понятие информационного графа этой РБД.

Информационный граф одной РБД — это граф $G = (V, \Gamma)$, в котором множество вершин V задает совокупность информационных узлов РБД, а многозначное отображение Γ задает связь между элементами множества V .

Выясняя это определение, можно отметить, что информационные ресурсы РБД — это территориально распределенные потребительские файлы данных, программные продукты, а так же отдельные базы данных. Информационный узел РБД состоит из тех информационных ресурсов РБД, которые принадлежат одному географическому месту, в котором расположена часть РБД или вся РБД.

Множество V называется множеством информационных узлов информационного графа G .

Предполагается, что один информационный узел содержит только одну модель данных. Отображение Γ определено так, что каждому информационному узлу сопоставляется множество всех информационных узлов, имеющих одинаковую модель данных с этим информационным узлом.

Определения отображения Γ дает возможность вывода ряд интересных свойств [5].

Пусть C_{m_i} ($i=1,2,\dots,n$) множество, которое содержит все информационные узлы одной модели данных m_i рассматриваемой РБД. C_{m_i} является i -той связанной компонентой соответствующего информационного графа $G=(V, \Gamma)$.

Предложение. Пусть $a_{m_i} \in C_{m_i}$ ($i=1,2,\dots,n$) один информационный узел в РБД. Множество C_{m_i} составлено из информационного узла a_{m_i} и всех информационных узлов, которых можно связать с узлом a_{m_i} при помощи некоторого пути.

Предложение. Множество C_{m_i} ($i=1,2,\dots,n$) генерирует подграф $G_{m_i} = (C_{m_i}, \Gamma_{m_i})$ информационного графа $G=(V, \Gamma)$ так что:

$$C_{m_i} \subseteq V$$

и

$$\Gamma_{m_i}(x) = \Gamma(x) \quad \text{для всех } x \in C_{m_i}$$

Легко можно доказать, что $\{C_{m_i}\}_{i=1,2,\dots,n}$ составляет разбиение множества V . Оказывается, что информационный граф одной однородной РБД является связанным графом.

Между информационными узлами информационного графа одной неоднородной РБД существуют два типа связи:

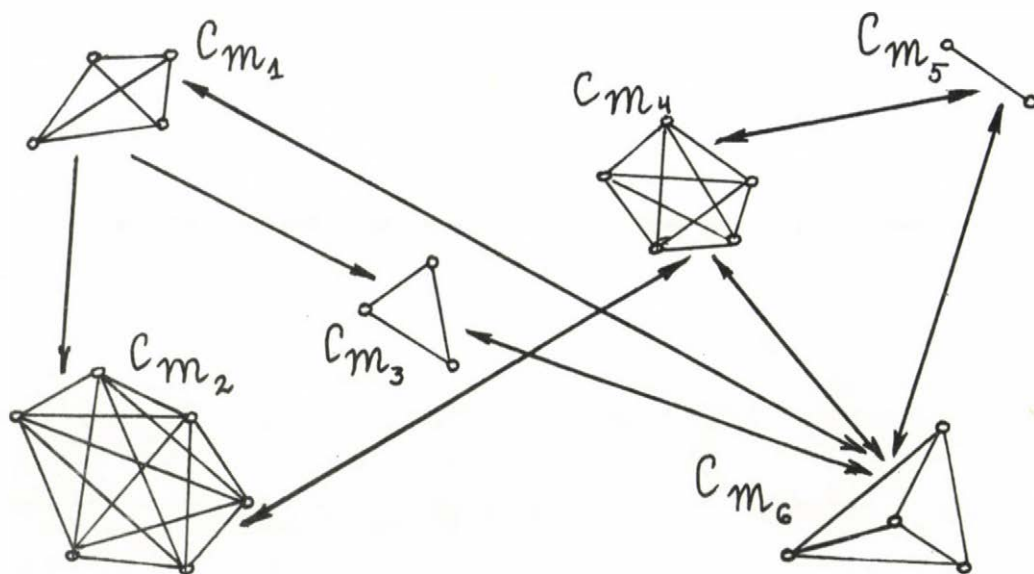
- связь между информационными узлами, имеющими одинаковую модель данных;
- связь между информационными узлами, имеющими разные модели данных.

Во втором случае возникает проблема трансформирования моделей данных, которая решается при помощи выбора глобальной модели данных (ГМД). Функциональный граф одной неоднородной

РБД отражает связь второго типа.

Функциональный граф - это граф $\mathcal{G} = (\mathcal{C}, \Phi)$, в котором множество вершин \mathcal{C} состоит из связанных компонент информационного графа $G = (V, \Gamma)$ этой РБД, а множество дуг Φ характеризует связь между элементами множества \mathcal{C} (фиг. 1).

Определим множество Φ так, что между любыми двумя компонентами C_{m_i}, C_{m_j} ($i, j = 1, 2, \dots, n; i \neq j$) можно найти хотя бы один путь. Это соответствует существенной характеристике одной РБД, что независимо от своего территориального местоположения, каждый потребитель имеет возможность пользоваться информацией, находящейся в каждом информационном узле.

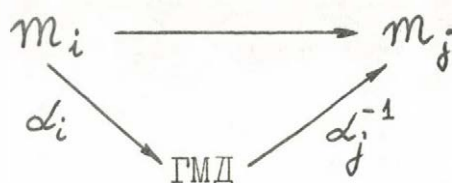


Фиг.1. Функциональный граф РБД с компонентами C_{m_i} ($i = 1, 2, \dots, 6$)

Если необходимо осуществлять трансформацию моделей данных $f: m_i \longrightarrow m_j$ для $i \neq j$ и $i, j = 1, \dots, n$ надо определить функции:

$$\alpha_i: m_i \longrightarrow \text{ГМД} \quad \text{и} \quad \alpha_j: m_j \longrightarrow \text{ГМД},$$

которые обратимы и такие, что диаграмма:



коммутативна, т.е.:

$$f = \alpha_j^{-1} \circ \alpha_i.$$

Узлы функционального графа составны, но они отличаются дискретным характером. Поэтому становится возможным менять алгебраические методы, чтобы описать их. В рассматриваемом случае оказываются подходящими некоторые элементы теории категорий.

Можно легко показать, что множество информационных узлов информационного графа G , составляющих одну связанную компоненту C_{m_i} ($i=1,2,\dots,n$), является категорией.

Из этого факта следует, что каждая вершина функционального графа $\mathcal{G} = (\mathcal{C}, \Phi)$ является категорией. Множество \mathcal{C} называется множеством категорий функционального графа \mathcal{G} .

Если в конкретном случае категория функционального графа $C_{m_i} \in \mathcal{C}$ ($i=1,2,\dots,n$) отличается так, что все операции, допустимые в языке манипулирования с данными модели m_i , являются ассоциативными, то категорию C_{m_i} можно рассматривать как категорию моноидов, вместе со всеми специфичными ей свойствами.

Некоторому потребителю РБД обычно необходимы только данные, определенных информационных узлов. Оказывается целесообразным ввести понятие "подкатегории" функционального графа \mathcal{G} .

Из организации одной вершины функционального графа \mathcal{G} как категория с объектами - информационные узлы, следует что можно определить понятие функтор между категориями функционального графа в соответствии с общей теорией категорий [1]. Можно доказать, что число отличных возможных функторов между категориями C_{m_i} и C_{m_j} ($\forall i, j=1,2,\dots,n; i \neq j$), принадлежащими множеству \mathcal{C} , равно числу разных путей между ни-

ми в функциональном графе $\mathcal{C} = (C, \Phi)$.

Доказательство следует непосредственно из того, что каждый функтор в \mathcal{C} интерпретируется при помощи возможной связи между соответствующими двумя категориями, изображенной некоторым элементом множества Φ или при помощи композиции таких элементов. Естественно можно определить и понятие "морфизма функторов" в графе \mathcal{C} .

То что рассматривалось до сих пор было необходимо, чтобы прийти к понятию об эквивалентности категорий из множества C , которое используется при исследовании трансформации моделей данных. Каждая трансформация модели m_i в модель m_j состоит из отдельных шагов:

- преобразование схем базой данных модели m_i в схемы базой данных модели m_j ;
- преобразование состояний базой данных модели m_i в состояния базой данных модели m_j ;
- преобразование операторов программ на языке манипулирования с данными модели m_i в операторы программ на языке манипулирования с данными модели m_j .

Чтобы трансформация моделей данных была корректна, нужно удовлетворить следующим требованиям:

- полная определимость;
- представимость;
- репродуктивность.

всех типов данных в РБД и соответствующих операторов над ними [4].

Определение. Модель данных m_i ($i=1,2,\dots,n$) эквивалентна модели данных m_j ($j=1,2,\dots,n$), если соответная категория C_{m_i} эквивалентна категории C_{m_j} .

Определение. Трансформация моделей данных $T : m_i \longrightarrow m_j$, которая трансформирует модель m_i в эквивалентную ей модель m_j , называется трансформацией эквивалентности.

Можно легко вывести, что если существует трансформация эквивалентности $T : m_i \longrightarrow m_j$, то тогда можно найти трансформацию моделей данных $U : m_j \longrightarrow m_i$ и два изоморфизма трансформаций $\Psi : 1_{m_j} \longrightarrow TU$ и $\Phi : 1_{m_i} \longrightarrow UT$,

так что выполнено условие:

$$T \cdot \psi = \psi \cdot T.$$

Это приводит к следующей теореме:

Теорема 1. Пусть $T : \mathcal{M}_i \longrightarrow \mathcal{M}_j$ трансформация моделей данных ($i, j = 1, 2, \dots, n; i \neq j$). Для того, чтобы T была трансформацией эквивалентности, необходимо и достаточно:

а) для каждой пары моделей данных $M_A, M_B \in \mathcal{M}_i; A, B \in \mathcal{C}_{\mathcal{M}_i}$ функция:

$$T(M_A, M_B) : \text{Hom}_{\mathcal{M}_i}(M_A, M_B) \longrightarrow \text{Hom}_{\mathcal{M}_j}(T(M_A), T(M_B))$$

является биекцией;

б) каждый объект $M_X \in \mathcal{M}_j$ для $X \in \mathcal{C}_{\mathcal{M}_j}$ должен быть изоморфным с объектом вида $T(M_E)$ для $M_E \in \mathcal{M}_i$ и $E \in \mathcal{C}_{\mathcal{M}_i}$.

Эта теорема доказывается при помощи следующего предложения:

Предложение. Пусть $T : \mathcal{M}_i \longrightarrow \mathcal{M}_j$ ($i \neq j; i, j = 1, \dots, n$) трансформация эквивалентности и $U : \mathcal{M}_j \longrightarrow \mathcal{M}_i$, $\psi : 1_{\mathcal{M}_i} \longrightarrow UT$ и $\psi : 1_{\mathcal{M}_j} \longrightarrow TU$ выполняют условия, упомянутые выше. Тогда верны следующие утверждения:

а) для каждой пары моделей данных $M_A, M_B \in \mathcal{M}_i$ соответствующих информационным узлам $A, B \in \mathcal{C}_{\mathcal{M}_i}$, функция:

$$T(M_A, M_B) : \text{Hom}_{\mathcal{M}_i}(M_A, M_B) \longrightarrow \text{Hom}_{\mathcal{M}_j}(T(M_A), T(M_B))$$

является биекцией;

б) трансформация U является сопряженной трансформации T и обратно;

в) выполнено условие: $U \cdot \psi = \psi \cdot U$.

Интересными являются и следующие следствия:

Следствие. Пусть $T : \mathcal{M}_i \longrightarrow \mathcal{M}_j$ трансформация эквивалентности. Если $U', U'' : \mathcal{M}_j \longrightarrow \mathcal{M}_i$ удовлетворяют условиям для трансформации U , описанным выше, то U' и U'' являются канонически изоморфными.

Следствие. Пусть $T : \mathcal{M}_i \longrightarrow \mathcal{M}_j$ трансформация эквивалентности. Если $U : \mathcal{M}_j \longrightarrow \mathcal{M}_i$ выполняет условия, даны выше для нее, то U является тоже трансформацией эквива-

лентности.

Это важно, когда рассматриваем двухстороннюю связь между вершинами функционального графа.

Предложение. Пусть $T: m_i \longrightarrow m_j$ и $U: m_j \longrightarrow m_k$ трансформации эквивалентности. Тогда существует трансформация моделей данных $V: m_i \longrightarrow m_k$, которая тоже будет трансформацией эквивалентности для $i, j, k = 1, 2, \dots, n$.

Доказательство.

а) Докажем, что для $\forall M_A, M_B \in m_i$ и для $\forall A, B \in C m_i$ отображение:

$$V(M_A, M_B): \text{Hom}_{m_i}(M_A, M_B) \longrightarrow \text{Hom}_{m_k}(V(M_A), V(M_B))$$

является биекцией.

Пусть $V = U \circ T$. Покажем, что это искомая трансформация.

Очевидно, что диаграмма:

$$\begin{array}{ccc} \text{Hom}_{m_i}(M_A, M_B) & \xrightarrow{U \circ T} & \text{Hom}_{m_k}(U(T(M_A)), U(T(M_B))) \\ & \searrow T & \nearrow U \\ & \text{Hom}_{m_j}(T(M_A), T(M_B)) & \end{array}$$

коммутативна. Тогда и диаграмма:

$$\begin{array}{ccc} \text{Hom}_{m_i}(M_A, M_B) & \xrightarrow{V} & \text{Hom}_{m_k}(U(T(M_A)), U(T(M_B))) \\ & \searrow T & \nearrow U \\ & \text{Hom}_{m_j}(T(M_A), T(M_B)) & \end{array}$$

коммутативна, потому что

$$V(M_A) = (U \circ T)(M_A) = U(T(M_A))$$

для $\forall M_A \in m_i$ и $\forall A \in C m_i$.

Так как T и U - трансформации эквивалентности, из условия необходимости теоремы 1. следует, что:

$$T(M_A, M_B) : \text{Hom}_{m_i}(M_A, M_B) \longrightarrow \text{Hom}_{m_j}(T(M_A), T(M_B))$$

и

$$U(T(M_A), T(M_B)) : \text{Hom}_{m_j}(T(M_A), T(M_B)) \longrightarrow \text{Hom}_{m_k}(U(T(M_A)), U(T(M_B)))$$

являются биекциями, т.е.:

$$\begin{aligned} V(M_A, M_B) &= (U \circ T)(M_A, M_B) : \text{Hom}_{m_i}(M_A, M_B) \longrightarrow \\ &\longrightarrow \text{Hom}_{m_k}((U \circ T)(M_A), (U \circ T)(M_B)) = \\ &= \text{Hom}_{m_k}(V(M_A), V(M_B)) \end{aligned}$$

является тоже биекцией.

б) Пусть $M_P \in m_k$ для $\forall P \in C_{m_k} (k=1, 2, \dots, n)$. Так как U - трансформация эквивалентности, из теоремы 1 следует, что существует $M_X \in m_j$ для $X \in C_{m_j}$, так что:

$$M_P = U(M_X).$$

Но и T - трансформация эквивалентности. Из теоремы 1 следует, что существует $M_A \in m_i$ для $A \in C_{m_i}$ и поэтому

$$M_X = T(M_A).$$

Тогда получается, что:

$$M_P = U(M_X) = U(T(M_A)) = (U \circ T)(M_A) = V(M_A).$$

Из а) и б) и теоремы 1 следует что V - трансформация эквивалентности.

Теорема 2. Если для каждого $i=1, 2, \dots, n$ существует трансформация эквивалентности $T : m_i \longrightarrow \text{ГМД}$, которая является обратимой, то для $\forall j=1, 2, \dots, n$ и $j \neq i$, существует трансформация моделей данных:

$$U_{ij} : m_i \longrightarrow m_j$$

которая определяет эквивалентность модели данных m_i с моделью данных m_j , которая обратима.

Из последнего предложения следует, что если существуют трансформации эквивалентности $T_i : m_i \longrightarrow \text{ГМД}$ и $T_j^{-1} : \text{ГМД} \longrightarrow m_j$, то можно составить композицию:

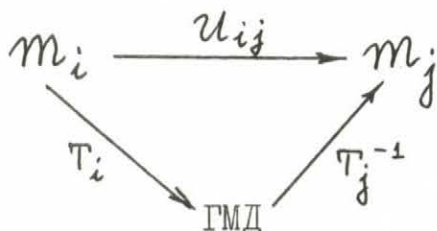
$$u_{ij} : T_j^{-1} \circ T_i,$$

которая будет трансформацией эквивалентности.

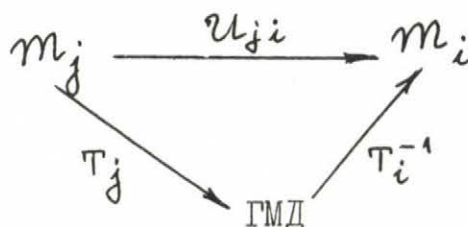
Наоборот, если возьмем $u_{ji} = (u_{ij})^{-1}$, тогда

$$u_{ji} = (T_j^{-1} \circ T_i)^{-1} = T_i^{-1} \circ T_j$$

является тоже трансформацией эквивалентности в обратном направлении. Это видно из следующих коммутативных диаграмм:



и



Следствие. В РБД, в которой $\forall i = 1, 2, \dots, n$ существует трансформация эквивалентности $T_i : m_i \longrightarrow \text{ГМД}$, которая обратима, все n модели данных эквивалентны между собой.

ЛИТЕРАТУРА:

1. Bucur, I., Algebra omologica, Editura Didactica si Pedagogica, Bucuresti, 1965.
2. Tsichritzis, D. and F. H. Lochovsky, Data Base Management Systems, Academic Press, Inc., New York, 1977.
3. Adiba, M., J.C. Chupin, R. Demolombe, G. Gardarin and J. Le Bihan, Issues in Distributed Data Base Management Systems: A Technical Overview, VLDB, Berlin 1977.
4. Kalinichenko, L.A., Data Models Transformation Method Based on Axiomatic Data Model Extension, VLDB, Berlin 1977.
5. Златарова, Ф., Върху някои проблеми на системите с разпределени информационни ресурси, Сборник от доклади на Школата по "Математични методи на информатиката", Варна, 1979.



ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ ПРИ ОРГАНИЗАЦИИ НАУЧНЫХ МЕРОПРИЯТИЙ

Румяна К. Киркова

Институт математики с ВЦ, БАН

При создании информационных систем основными являются анализ и синтез циркулирующей в них информации, так как их функционирование связано с формированием, регистрацией, сбором, обработкой и хранением информации, адекватно отражающей состояние определенных объектов в процессе их развития.

При организации одного конгресса или симпозиума (все эти мероприятия будем называть научное мероприятие – НМ) основными задачами являются:

- сообщения о НМ – научная тематика, время проведения, место проведения, условия на участие;
- корреспондентная связь с участниками НМ;
- составление научной программы НМ с учетом научных направлений, разбиением по секциям, залам, дням и т.д.;
- обеспечение соответствующих условий по пребыванию – проезд, гостиница, регистрация, отъезд и т.д.;
- участие в дополнительных (параллельном основному) заседаниях, встречах, дискуссиях и пр. с одной стороны, а с другой – ознакомление с культурными, природными и историческими достопримечательностями страны.

Обычно для каждого НМ формируется организационный комитет (ОК). Этот ОК проводит все вышеупомянутые задачи. При решении этих задач очень удобным инструментом является информационная система (ИС). Объектом информационного банка такой ИС является каждый участник НМ. Данные для каждого участника можно сгруппировать как следует:

- личные данные: имя, отчество и фамилия; научное звание и/или научная степень; научная организация (ад-

рес и наименование); адрес для корреспонденции (государство, город и т.д.);

- данные об участии в научной программе: с/без научного сообщения; вид сообщения (доклад по приглашению, пленарный доклад, научное сообщение, в качестве соавтора и пр.); наименование, язык, научное направление; имена всех авторов.

Если предусматриваются несколько сообщений для каждого из них необходимо сохранять все эти характеристики;

- данные по пребыванию: число сопровождающих лиц; предпочитаемый тип гостиницы и комнаты; предполагаемое время приезда (дата, час, вид транспорт); предполагаемое время отъезда (дата, час, вид транспорт); финансовые условия на участие (по приглашению, за свой счет, за счет эквивалентного безвалютного обмена и пр.);
- дополнительные требования: например участие в параллельных заседаниях, встречах, дискуссиях, как и в экскурсиях с целью ознакомления с культурными, природными или историческими достопримечательностями места проведения НМ.

Конечно для каждое НМ части этой информации можно не накапливать в случае когда перед соответствующему ОК не стоят одних или других задач (например нет параллельных мероприятий; ОК не заботится о гостиницах и т.д.).

Сбор этой информации проводится в разных этапах организации НМ — так например основные личные данные получают после ответа первому сообщению; данные об участии в научной программе — после ответа второму сообщению; а данные по пребыванию содержатся обычно в телеграмме перед приездом участника. Иногда во время проведения НМ часть информации актуализируется (например время отъезда, адрес, участие в дискуссиях или экскурсиях и т.д.).

На базе накопленной информации из информационной системой после соответствующей обработки можно получать разные выходные материалы в разных направлениях:

- для корреспонденции: списки участников с их адресами (эventуально эти списки могут быть упорядочены по фамилиям или государствам); список участников, которые уже ответили / еще не ответили на первом / втором сообщении;
- для составления научной программы: списки докладов и сообщений по отдельным направлениям или секциям; список всех пленарных докладов или всех докладов по приглашению; список всех авторов; справка о числе и типах докладов и сообщения каждой секции; версия научной программы по залам, дням, часам и пр.;
- для регистрации: список участников с информацией о времени приезда, финансовых условиях, гостиницах, сопровождающих лиц, дополнительных требованиях (эventуально эти списки можно упорядочить любым параметром - государство, дата и час приезда, тип гостиницы и пр.).

Создание информационной системы для обслуживания НМ можно реализовать разными способами. Один из самых удачным и удобным является реализация на основе некоторой системы управления базами данных (СУБД), так как известно что каждая СУБД предоставляет основные стандартные средства ввода, актуализации, хранения, обработки и вывода информации. Кроме того время реализации информационной системы на основе СУБД резко сокращается.

В Институте математики с Вычислительным центром Болгарской АН - сектор "Основы кибернетики и теория управления", с 1973 года проводятся исследования, реализации и эксплуатации информационных систем такого типа. Некоторые из них [1] были созданы для ЭВМ МИНСК-32, другие - для ЕС ЭВМ. Учитывая преимущества применения СУБД при создания ИС для НМ в 1978

году под руководством доц. Д. М. Добревым начала разработка типовой информационной системы для автоматизирования некоторых процессов при организации научного мероприятия под названием КИСИМ на основе СУБД БИСЕС. Основные принципы, архитектура и возможности СУБД БИСЕС описаны в [2] и [3].

Типовая информационная система КИСИМ была использована для обслуживания ряд конгрессов, конференции и симпозиумов, организованных Единым центром математики и механики Болгарской АН в периоде 1978 - 1981 г.г. Применение СУБД БИСЕС дает возможность быстро и легко расширять набор выходных материалов новыми списками или справками. Широкие возможности ввода и актуализации информации обеспечивают связь с разными форматами используемых форм на регистрации участия в НМ. Для сохранения многих из характеристик принята сжатая (компримированная) форма (шифр государства, шифр секции, шифр научного направления, шифр транспорта приезда/отъезда, шифр типа научного сообщения, шифр параллельного мероприятия и т.д.). Используя подходящие классификаторы после обработки в выходных материалах все шифры заменяются соответствующими текстами. При этом есть возможность выбора языка для объяснительных текстов (русский, английский и пр.).

На базе общей научной области (математика) в результате эксплуатации уже создан единый информационный банк (ЕИБ) всех участников и всех научных мероприятий. Кроме вышеуказанных характеристик в этом ЕИБ для каждого участника сохраняются и признаки отдельных НМ, в которых он принимал участие как и его научные интересы. Таким образом для каждого нового НМ соответствующий организационный комитет имеет возможность сразу пользоваться услугами вычислительных машин и информационной системы (например отпечатать адреса эвентуальных участников и разослать первое сообщение; получить список интересующихся специалистов в определенной научной области и т.д.).

ЛИТЕРАТУРА

- 1 Киркова, Р., Петрова, В., Ескенази, А. - Система для организации научных конгрессов и симпозиумов с применением ЭВМ (на болгарском языке), София, журнал "Системы и управления", кн. 1/1976, с.с. 37-41
- 2 Труды Конференции по системам информационного обслуживания коллективов профессионально-связанных потребителей, Варна, май 1977, с.с. 170-197
- 3 Добрев, Д.М., Денев, И.Д., Киркова, Р.К., Ескенази, А.М., Швертнер, И.Б., Парванов, П.А. - Архитектура и основные принципы СУБД БИСЕС, Москва, сб. Проблемы информатики (в печати)

С О К Р А Т
СИСТЕМА ДЛЯ ОПЕРАТИВНОГО КОНТРОЛЯ НАД
РЕАЛИЗАЦИЕЙ И ОТЧЕТОМ ЗАГРАНИЧНЫХ СЛУ-
ЖЕБНЫХ КОМАНДИРОВОК

Румяна К. Киркова

Институт математики с БЦ Болгарской АН

В последнее время в различных областях деятельности человека широкое применение получили так называемые автоматизированные системы обработки данных, основанные на использовании ЭВМ. Это предъявляет повышенные требования к их параметрам, прежде всего к производительности, доступности для специалистов различных профилей и квалификации, степени автоматизации процессов запуска и управления вычислительным процессом и т.д.

На пути создания систем обработки данных возникает целый ряд взаимосвязанных проблем практического и теоретического характера, от успешного решения которых в итоге зависит эффективность систем обработки данных. Среди них такие, как автоматизация процесса синтеза алгоритмов и программ обработки данных, оптимизация интерфейсов средств обработки данных и т.д.

Эффективность автоматизированных систем обработки данных существенным образом зависит и от методов построения и организации их системного и проблемного математического обеспечения. Именно поэтому в настоящее время уделяется большое внимание поиску новых эффективных формальных методов анализа и синтеза программного обеспечения систем обработки данных.

На фоне постоянно возрастающего потока информации, необходимой для принятия обоснованного управленческого решения, особое значение приобретает поиск эффективных средств автоматизации процессов документирования учрежденческой дея-

тельности. Значительный научный и практический интерес представляет проблема разработки, исследования и внедрения различных элементов программного обеспечения ЭВМ и их комплексов.

Центральной проблемой создания любого информационного банка является выбор объекта и его информационную модель, от обоснованности, точности и достоверности которой во многом зависит эффективность системы обработки данных.

Другой из актуальных задач, представляющих интерес при решении этой проблемой, является задача создания средств автоматизации процесса обработки данных – сложного и трудоемкого этапа исследования в различных областях науки и техники.

Обзор достижений в этом направлении показывает, что в последнее время применяется хорошо зарекомендовавший себя метод универсальных средств – системы управления базами данных (СУБД) (см. например [1], [2]).

Пользуясь этим аппаратом в Институте математики с Вычислительным центром Болгарской АН – сектор "Основы кибернетики и теории управления", под руководством доц. Д. М. Добревым с 1973 года были реализованы ряд автоматизированных систем обработки данных разного типа и разной степени сложности построения информационного банка. Две из этих систем описаны в [3] и [4].

Основой программного обеспечения является СУБД БИСЕС (Базовая Информационная Система для вычислительных машин Единой Системы), которая обеспечивает соответствующие универсальные средства генерирования информационного банка; сбора, обновления, хранения и обработки информации; вывод результатов обработки в заданном стандарте. Основные принципы, архитектура и возможности СУБД описаны например в [5] и [6].

Система для оперативного контроля над реализацией и отчетом заграничных служебных командировок (ЗСК) СОКРАТ также реализована на базе программного продукта СУБД БИСЕС.

Чтобы охарактеризировать одну ЗСК необходимо иметь следующие данные: цель ЗСК (например для участия в международной конференции или симпозиуме; для работы по совместной теме; на стажировку и т.д.); страна; срок пребывания; финансовые условия (по эквивалентному безвалютному обмену, за счет одной или другой научной организации и т.п.); время (начало и конец) пребывания; место пребывания (например город и научная организация); дата отъезда и дата приезда; предоставленные отчетные материалы и т.д.

С другой стороны надо располагать данными о сотруднике, который реализовал эту ЗСК. Основными данными являются: имя, отчество и фамилия; занимаемая должность; научное звание; научная степень; организация; подразделение этой организации (лаборатория, отдел, секция и т.д.).

Сразу видно, что динамической частью информационного банка являются данные об отдельных ЗСК. Так называемые личные данные сотрудника в принципе можно считать постоянными. Поэтому информационный банк системы о ЗСК можно рассматривать как специализированный файл, связанным системой указателей общей базе системы кадров данной научной организации. Таким образом вводимая информация резко уменьшается по своему объему, а связь с общей базой системы кадров увеличивает возможность получения ряд разнообразных выходов.

Так как в Едином центре математики и механики Болгарской АН была реализована и система кадров, используя единый классификатор научных специалистов БАН, были созданы предпосылки реализовать информационный банк системы СОКРАТ вышеупомянутым способом в виде специализированного файла с указателями на личные данные.

Ответственные для актуальности информации системы СОКРАТ являются сотрудниками отдела "Международное сотрудничество". После утверждения данной ЗСК соответствующим приказом вышестоящих органов они стандартным форматом задают в системе необходимую информацию для ЗСК, как и указатель на лич-

ные данные определенного сотрудника. После реализации запись о ЗСК актуализируется новыми данными (например дата отъезда и приезда, актуальный срок пребывания, признаки о предоставленных отчетных и других материалах и т.п.).

В настоящее время в этом файле накоплена информация о всех реализованных с 1976 года сотрудниками Единого центра математики и механики ЗСК.

В определенных периодах или по требованиям из системы получают необходимые списки и справки - все результаты оформлены согласно принятым стандартам. Используя стандартные возможности базового продукта - СУБД БИСЛО, этот набор выходных результатов непрерывно расширяется, как в отношении объема информации, так и по направлениям - по годам реализации, по странам, по сотрудникам, по целям ЗСК, по типам ЗСК и т.д.

Так как в каждой записи задаются и признаки о предоставлении отчетных материалов из системы СОКРАТ периодически выводится список тех ЗСК, о которых нет отчетов - этот список оформлен в виде адресованных сообщений к соответствующим сотрудникам научной организации. Этим контролируется и управляется системой отчетности ЗСК.

Кроме сотрудников отдела "Международное сотрудничество" пользователями информации системы СОКРАТ являются и другие инстанции - Руководство Единого центра, Научный секретариат Единого центра, Научный архив и т.д. Они высоко оценивают эффективность и оперативность реализованной системы.

В настоящее время проводятся некоторые работы по расширению области применения системы СОКРАТ в нескольких направлениях. Первое направление это включить в информационном банке и в соответствующих обработках план о ЗСК. Таким способом будет возможно контролировать и учитывать еще и признаки о неосуществленных ЗСК.

Второе направление связано с созданием в рамках

Болгарской АН единой системы кадров. Это создает объективные предпосылки расширить информационный банк системы СОКРАТ для всех подразделений БАН. Тогда на разных уровнях создается возможность получения необходимой для принятия обоснованного управленческого решения информации, как и автоматизировать процессы административных деятельностей.

ЛИТЕРАТУРА

- 1 Дейт, К. - Введение в системы баз данных, Москва, Издательство Наука, 1980
- 2 Мартин, Дж. - Организация баз данных в вычислительных системах, Москва, Издательство Мир, 1978
- 3 Добрев, Д.М., Киркова, Р.К., Парванов, П.А. - Система планирования и учета вычислительных ресурсов (в настоящем сборнике)
- 4 Киркова, Р.К. - Применение информационных систем при организации научных мероприятий (в настоящем сборнике)
- 5 Труды Конференции по системам информационного обслуживания коллективов профессионально-связанных потребителей, Варна, май 1977, с.с. 170-197
- 6 Добрев, Д.М., Денев, И.Д., Киркова, Р.К., Ескенази, А.М., Швертнер, И.В., Парванов, П.А. - Архитектура и основные принципы СУБД БИСЕС, Москва, сб. Проблемы информатики (в печати)

ПОДХОД К ПОСТРОЕНИЮ МАТЕМАТИЧЕСКОЙ ТЕОРИИ ДЛЯ КЛАССОВ ИНФОРМАЦИОННЫХ СИСТЕМ

А.И. Кондратьев

Институт кибернетики АН УССР

ВВЕДЕНИЕ

Необходимость создания общей математической теории АИС продиктована в первую очередь требованиями автоматизации процессов проектирования АИС. Следует отметить, что современное состояние работ по созданию общей математической теории АИС является неудовлетворительным. Накоплен значительный опыт по созданию практических систем и получен ряд важных теоретических результатов, среди которых следует, в первую очередь, отметить модели Э.Ф.Кодда /1/. В то же время общая математическая теория АИС отсутствует и неясны тенденции и пути ее развития. При становлении этой теории следует придерживаться следующих трех этапов ее формирования:

1. Формализуются все компоненты и процессы АИС; вводится математическая модель для конкретных классов АИС.
2. Вводится общая модель для всех классов АИС.
3. Рассматривается пространство моделей; разрабатывается единый математический аппарат; целиком изучаются все свойства АИС.

Современное состояние АИС позволяет выделить 4 основных класса АИС /2/: информационно-поисковые; информационно-справочные; информационно-логические; информационно-распознающие.

Обычно для АИС рассматриваются два процесса: проектирование и функционирование, описываемые с помощью директивного и информативного контуров /3/.

При создании общей математической теории АИС необходимо ориентироваться: во-первых, на различный характер информационных объектов; во-вторых, на разнообразие сферы применения; в-третьих, на противоречивости и конфликтность интересов пользователей, а также процессов проектирования и функционирования АИС. Учет перечисленных аспектов позволяет использовать при

создании общей математической теории АИС нормативную теорию игр /4,5/.

В докладе рассматриваются некоторые вопросы трех основных этапов создания общей математической теории АИС для информационно-распознающих систем.

ИСХОДНЫЕ ПОНЯТИЯ

Для описания АИС используется понятие общей схемы G , содержащей контур информации F и контур алгоритмов D .

Контур информации F является формализацией понятия информативный контур /3/ и содержит H взаимосвязанных типов описаний объектов Λ /4/. Обозначим каждое из описаний, используемое на шаге h n -прохода G общей схемы G через $F_{nh} \in \{F_{nh}\}$, $1 \leq h \leq H$, $1 \leq n \leq N$. Алгоритм формирования перехода от F_{h-1} к F_h обозначим через $e_h \in \{e_h\}$, $1 \leq h \leq H$.

Контур алгоритмов D является формализацией понятия директивный контур /3/ и содержит множества $\{e_h\}$, $1 \leq h \leq H$. В том случае, если решение и переход от F_{h-1} к F_h производится на основе накопленного опыта, то алгоритм e_h является алгоритмом классификации /4,6/ и записывается таким образом: необходимо по описанию F_{h-1} на основе множества начальных информации о компонентах разбиения F_h определить к каким из компонент F_h относится F_{h-1} . Пусть $\{F_{h-1}\}^0$ - множество исследованных описаний, для которых известно соответствующее множество описаний на следующем шаге - $\{F_h\}^0$. Это множество может рассматриваться как априорно известное или специальным образом построенное. Обозначим через $\{\pi_j(F_{h-1})\}$ - классы понятий, составляющие компоненты описания F_{h-1} . Через $|\{\pi_j(F_{h-1})\}|$ обозначим число классов понятий в F_{h-1} , а через $|\{F_h\}^0|$ - размерность множества $\{F_h\}^0$. Разобьем каждый из алгоритмов e_h на произведение оператора α_h на решающее правило алгоритмов γ_h . Целесообразно рассматривать

X_h как оператор, который переводит следующее множество
 $\langle F_{h-1} \in \{F_{h-1}\}, \{F_{h-1}\}^0, \{F_h\}^0 \rangle$ в числовую матрицу, $\| \xi_{ij} \|$,
 $1 \leq i \leq |\{F_{h-1}\}^0|$; $1 \leq j \leq |\{X_j(F_h)\}|$.

Решающее правило Y_h переводит числовую матрицу $\| \xi_{ij} \|$ в матрицу с компонентами $\| \tau_{ij} \|$, где $\tau_{ij} = 0, 1$. В этом случае

алгоритм e_h будем называть сильно корректным. Корректным алгоритмом будем называть алгоритм, находящий корректное решение прямой и обратной задач нормативной теории игр /4/. Обозначим через

Δ тот факт, что алгоритм e_h отказался выполнить предписанное ему преобразование. Алгоритм e_h , для которого $\tau_{ij} \in \{0, 1, \Delta\}$ назовем сильно некорректным. Некорректным алгоритмом будем называть алгоритм, находящий некорректное решение прямой и обратной задач нормативной теории игр /4/.

Определение 1. Компонентами общей схемы назовем компоненты контура информации и компоненты контура алгоритмов.

Определение 2. Общую схему назовем инвариантной относительно своих компонент, если за конечное число переходов решение прямой и обратной задач нормативной теории игр можно сделать корректным.

Определение 3. Инвариантной общей схемой назовем общую схему, инвариантную относительно своих компонент и числа переходов. Общая схема описывается следующей моделью при $1 \leq h \leq H$

$$\begin{aligned} O &= \langle \{F_h\}, \{e_h\}, \{G\} \rangle \\ F &= \{F_h\}, \quad D = \{e_h\}. \end{aligned} \quad (I)$$

Можно показать, что в модель /I/ укладываются все 4 класса АИС, их характеристики и процессы при условии фиксации компонент этой модели.

ВОПРОСЫ ТЕОРЕТИКО-ИГРОВОГО МОДЕЛИРОВАНИЯ ИНФОРМАЦИОННО-РАСПОЗНАЮЩИХ СИСТЕМ

Рассмотрим в условиях работ /2-6/ задачу распознавания χ^c в стандартной постановке. Заданы совокупность допустимых объектов $\{\Lambda\}$ и разбиение множества понятий $\{U\}$ на классы $\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_m$. Кроме того, задана информация $\{S_0\}$ о классах $\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_m$. Необходимо $\forall \Lambda \in \{\Lambda\}$ на основе $\{S_0\}$ и информации S о Λ определить к каким из классов $\mathcal{K}_1, \mathcal{K}_2, \dots, \mathcal{K}_m$ принадлежит Λ .

В тех же предположениях, что и выше, целесообразно ввести χ^H задачу распознавания в нестандартной постановке, когда информация о Λ задана в противоречивой форме. Обозначим этот факт через $S^H(\Lambda)$. (тот случай в настоящее время в теории распознавания не рассматривается, хотя он совершенно реален и естественен, например, объект "противится", не дает возможности прямо или косвенно получить достоверную информацию и создает различные помехи.

Задачу распознавания будем рассматривать как антагонистическую игру Γ_P с Природой. Под игроком II будем понимать лицо, принимающее решение, а под игроком I - Природу. Стратегиями II являются алгоритмы e_x , множество которых составляет $\{e_x\}$. Стратегии Природы - это совокупность информационных описаний объектов $\{S(\Lambda)\}$. Ситуация в игре Γ_P образуется в результате выбора каждым из игроков своей той или иной стратегии. Обозначим ситуацию в игре Γ_P через $c = \{S(\Lambda), e_x\}$. Среди стратегий $\{e_x\}$ есть стратегии составления стандартной информации S_0 и, кроме того $S_0 \in \{S(\Lambda)\}$.

Определена функция выигрыша \mathcal{H} , заданная на множестве ситуаций $c \in \mathcal{C}$.

Определение 4. Теоретико-игровой моделью задачи распознавания назовем совокупность

$$\Gamma_P = \langle \{S(\Lambda)\}, \{e_x\}, \mathcal{H} \rangle$$

Процесс игры Γ_P рассматривается как выбор игроками I и II своих стратегий из $\{S(\lambda)\}$ и $\{e_x\}$ и полученные игроком I от игрока II его выигрыша $H(c)$.

Свяжем функцию H с функционалом $\varphi(e_x)$ [6].

$$\begin{aligned} \text{Положим } H(c) &= H(S(\lambda), e_x) = \\ &= H_{S(\lambda)}(e_x) = [1 - \varphi(e_x)] S(\lambda) \end{aligned}$$

Оптимальным поведением игроков в Γ_P принято считать такое их поведение, при котором выполнено:

$$\max_{\{S(\lambda)\}} \min_{\{e_x\}} H(S(\lambda), e_x) = \min_{\{e_x\}} \max_{\{S(\lambda)\}} H(S(\lambda), e_x).$$

Стратегии Природы $\{S(\lambda)\}$ фактически представляют собой деревья понятий, заданные на множестве понятий $\{U\}$ [4]. Это означает, что следует рассматривать не множество описаний $\{S(\lambda)\}$, а множество деревьев \mathcal{G} . Заметим, что в АИС в настоящее время известно три способа задания дерева \mathcal{G} :

- сетевой;
- иерархический;
- реляционный.

Теорема 1. В игре Γ_P существует ситуация равновесия, если $\{e_x\}$ - линейное векторное пространство, замкнутое относительно алгебраических операций (алгебры и σ -алгебры), а $\{\mathcal{G}\}$ компактно.

Теорема 2. Точность задачи распознавания определяется как $(1 - 2\varepsilon)$ для Γ_P , где за принцип оптимальности игроков принято стремление к ε - равновесным ситуациям.

Теорема 3. Ситуация равновесия в игре Γ_P задает сильно корректный алгоритм распознавания и оптимальное представление распознаваемой информации.

ЗАДАЧИ, РЕШАЕМЫЕ В ИНФОРМАЦИОННО-РАСПОЗНАВАЮЩИХ СИСТЕМАХ (ИРС)

Определение 5. Под ИРС понимается следующая совокупность:

$$\mathcal{R} = \langle L, \{W\}, \{Z\} \rangle$$

L - имеющаяся информация, информационное множество;

$\{Z\}$ - множество решаемых задач содержательной обработки информации;

$\{W\}$ - множество алгоритмов решения задач $\{Z\}$.

При этом $L \geq \bigcap_{n=1}^N \{F_n\}$, $\{W\} \geq \bigcap_{n=1}^N \{E_n\}$

Пусть даны множество задач $\{Z\}$, множество информации $\{I\} \subseteq L$, множество алгоритмов $\{W\}$, множество предикатов $\{P_r\}$, множество объектов $\{\Lambda\}$. Рассматриваются следующие задачи в $\{Z\}$.

Задача Z : построить алгоритм W , который по информации I для объектов $\{\Lambda\} = \{\Lambda_1, \dots, \Lambda_n\}$ вычисляет свойства, предикаты $\{P_r\} = \{P_{r_1}, \dots, P_{r_e}\}$. Задача Z в таком виде описана Д.И. Журавлевым [6].

Z_1 - Задачи распознавания, классификации, диагностики.

$\{\Lambda\} = \bigcup_{i=1}^e K_i$, $\{K_i\}$ - классы, возможно пересекающиеся, $P_{r_j}(\Lambda) - " \Lambda \in K_j "$, $1 \leq j \leq e$.

Z_1 рассматривалась Д.И. Журавлевым для стандартной информации I [6].

Z_2 - Задачи формирования понятий, классов, образов.

$\{\Lambda\} \cap \{U\} = \emptyset$, $\{U\}$ - множество понятий, образов.

$P_{r_j}(\Lambda) - " \Lambda \rightarrow U_j "$, $1 \leq j \leq e$.

Z_3 - Задачи установления связей между понятиями, классами понятий.

В условиях задачи Z_3 пусть $\{U^\alpha\}$ и $\{U^\beta\}$ - два подмножества понятий в $\{U\}$, причем возможно $\{U^\alpha\} \cap \{U^\beta\} \neq \emptyset$

$\alpha, \beta \in (1, \dots, n)$. Для понятий $U_i^\alpha \in \{U^\alpha\}$ необходимо вычислить свойства, предикаты

$$P_{rj}(U_i^\alpha) - "U_i^\alpha \rightarrow U_j^\beta", U_i^\alpha \in \{U^\alpha\}, U_j^\beta \in \{U^\beta\}, 1 \leq i \leq n, 1 \leq j \leq \ell.$$

¹7₄ - Задачи дискретного прогнозирования.

В условиях ¹7₂ $U_j = [a_j, b_j] \subset R^1$,

Λ - объект, P_{rj} - его характеристики, $1 \leq j \leq \ell$

¹7₅ - Задачи процессионного прогнозирования.

В условиях ¹7₄ Λ - процесс, P_{rj} - его характеристики,

$1 \leq j \leq \ell$.

¹7₆ - ¹7₉ - основные задачи теории игр.

¹7₆ - Задачи формирования игр.

$\{\Gamma\} = \{\Gamma_1, \Gamma_2, \dots, \Gamma_\ell\}$ - классы игр.

$$P_{rj}(\Lambda_i) - "\Lambda_i \rightarrow \Gamma_j", 1 \leq i \leq n, 1 \leq j \leq \ell.$$

¹7₇ - Задачи формирования принципов оптимальности.

$\{\varphi\} = \{\varphi_1, \varphi_2, \dots, \varphi_\ell\}$ - принципы оптимальности

$$P_{rj}(\Gamma_i) - "\Gamma_i \rightarrow \varphi_j", 1 \leq i \leq n, 1 \leq j \leq \ell$$

¹7₈ - Задачи проверки реализуемости принципов оптимальности.

$\{S\} = \{S_1, S_2, \dots, S_n\}$ - ситуации в игре Γ .

$$P_{rj}(S_i) - "\varphi(S_i)", 1 \leq j \leq \ell, 1 \leq i \leq n.$$

¹7₉ - Задачи нахождения оптимального решения.

В условиях задачи ¹7₈

$$P_{rj}(S_i) = 1, 1 \leq j \leq \ell, 1 \leq i \leq n.$$

¹7₁₀ - Задачи дискретной оптимизации с опытом.

$$P_{rj}(x) - "x \in X_j", 1 \leq j \leq \ell-1$$

$$P_{r\ell}(x) - "x = \text{extr } \varphi".$$

¹ 7₁₁ - Задачи дискретной многокритериальной оптимизации с опытом.

$$Pr_j(x_i) - "x_i = extr \varphi_j", 1 \leq i \leq n, 1 \leq j \leq \ell$$

¹ 7₁₂ - Задачи дискретного анализа.

$\{\chi\} = \{\chi_1, \chi_2, \dots, \chi_n\}$ - реализации в математической модели Γ / в том числе и игре/, а $\{\varphi\}$ - принципы оптимальности.

$$Pr_j(\chi_i) - "\chi_i \rightarrow \varphi_j", 1 \leq i \leq n, 1 \leq j \leq \ell.$$

¹ 7₁₃ - Задачи дискретного синтеза.

$\{\Lambda\}$ - компоненты, $\{\Gamma\}$ - синтезируемые системы /в том числе и игры/.

$$Pr_t(\Lambda_i) - "\Lambda_i \rightarrow \Gamma_t",$$

$$Pr^t = \bigwedge_{i=1}^n Pr_t(\Lambda_i) = 1, 1 \leq i \leq n, 1 \leq t \leq \ell$$

¹ 7₁₄ - Дискретный аналог задач принятия решений. Рассматривались Д.И. Журавлевым /6/.

Задачи ¹7₂ - ¹7₁₃ в таком виде описаны впервые.

Задача ²7: построить алгоритм W , который по информации I среди объектов $\{\Lambda\}$ указывает объект, обладающий свойствами

² 7₁ - Задачи дискретной оптимизации с опытом /прямые/.

$$Pr_j(x) - "x \in X_j", 1 \leq j \leq \ell-1$$

$$Pr_\ell(x) - "x = extr \varphi",$$

² 7₂ - Задачи дискретной многокритериальной оптимизации с опытом /прямые/.

$$Pr_j(\Lambda) - "\Lambda = extr \varphi_j", 1 \leq j \leq \ell$$

Задача ³7 : построить алгоритм W , который по информации I для объектов $\{A\}$ строит математическую модель Γ , находит для Γ , в чем заключаются оптимальные свойства $\{Pr\}$ объектов $\{A\}$ и указывает среди $\{A\}$ объект, обладающий этими свойствами.

³7₁ - Задачи теории игр.

$\{A\}$ - ситуации, Γ - игра, $\{Pr\}$ - характеристики принципа оптимальности φ , который может быть неизвестным.

³7₂ - Информационные задачи исследования операций.

φ в этом случае известен

Пусть φ - критерий оценки качества решения задачи ⁱ7 ($1 \leq i \leq 3$)

Задача ⁴7 : построить алгоритм W , который среди информации $\{I\}$ для задачи ⁱ7 ($1 \leq i \leq 3$) выбирает такую, на основе которой задача ⁱ7 решается наилучшим образом, т.е. достигается $\text{extr}_{I \in \{I\}} \varphi(\supset 7)$, $1 \leq i \leq 3$.

Задачи ²7 - ⁴7 в таком виде описаны впервые.

ОБЩАЯ СХЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧ В ИРС

Для решения задач $\{7\}$ разработана следующая схема.

1. Задание опыта и вида информации, задачи.
 2. Построение множества эвристических алгоритмов.
 3. Задание классов, решений, информационных векторов.
 4. Построение теоретико-игровой модели решения задачи $\supset 7$ в виде $\supset 7$
 5. Принцип, условие, критерий оптимальности в $\supset 7$.
 6. Оптимальное решение в $\supset 7$.
 7. Корректное решение задачи $\supset 7$.
 8. Внесение оптимальной и корректной информации в опыт, изменений, дополнений опыта, вида информации и переход к этапу 2.
- Покажем, как в общую схему укладываются алгоритмы решения задач $\supset 7$.

Описание об- щей схемы для решения задач {7}	Алгоритм решения задачи 17	Алгоритм решения задачи 27	Алгоритм решения задачи 7	Алгоритм решения задачи 7
1. Задание опыта и ин- формации, задачи	1. Выбор вида информации и описания (стандарт- ные)	1. Фиксация опыта	1. Фиксация опыта	1. Фиксация задачи и алгоритма
2. Построение множества эвристиче- ских алго- ритмов	2. Выбор эв- ристическо- го алгорит- ма	2. Выбор эв- ристическо- го алгорит- ма	2. Выбор эв- ристическо- го алгорит- ма	2. Выбор эв- ристичес- кого алго- ритма
3. Задание классов, ре- шений, инфор- мационных векторов	3. Информа- ционный вектор	3. Задание классов, ре- шений, пере- бор в клас- се	3. Задание классов, решений, перебор в классе	3. Информа- ционный вектор
4. Построе- ние теоре- тико-игро- вой модели решения за- дачи 7 в виде $\Gamma(7)$	4. Построе- ние модели алгоритмов	4. Построе- ние модели алгоритмов	4. Построе- ние модели алгоритмов	4. Построе- ние моде- ли инфор- маций
5. Принцип, условие, критерий оптималь- ности в $\Gamma(7)$	5. Задание критерия качества алгоритма при фикси- рованном опыте	5. Задание критерия оп- тимальности при фикси- рованном опыте	5. Задание критерия оптималь- ности при фиксиро- ванном опыте	5. Задание критерия качества информа- ции при фиксиро- ванном алгоритме и задаче

6. Оптимальное решение в $\Gamma(\gamma)$	6. Поиск оптимального алгоритма, доставляющего экстремум функционалу	6. Поиск оптимального алгоритма, доставляющего экстремум функционалу	6. Поиск оптимального алгоритма, доставляющего экстремум функционалу	6. Поиск оптимальной информации, доставляющей экстремум функционалу
---	--	--	--	---

7. Корректное решение задачи γ	7. Корректное решение задачи γ^1	7. Корректное решение задачи γ^2	7. Корректное решение задачи γ^3	7. Корректное решение задачи γ^4
---------------------------------------	---	---	---	---

8. Внесение оптимальной и корректной информации в опыт, изменений, дополнений опыта, вида информации и переход к этапу 2	8. Внесение оптимального решения в опыт, пополнение опыта и решение γ^1 , и переход к этапу 2	8. Внесение оптимального решения в опыт, пополнение опыта и решение γ^2 , переход к этапу 2	8. Внесение оптимального решения в опыт, пополнение опыта и решение γ^3 , переход к этапу 2	8. Внесение оптимальной информации в опыт, решение γ^4 , $1 \leq i \leq 3$ и переход к этапу 2
--	--	--	--	---

Разработанная идеология, аппарат и алгоритмы используются для решения практических задач по управлению функциональным и тактико-техническим состоянием спортсменов в спортивных играх и видах единоборств.

Л И Т Е Р А Т У Р А

1. Codd E.F. *A Relational Model of Data for Large Shared Data Banks.* - "Communication of the ACM", June, 1970, v. 6, n. 13.
2. Глушков В.М., Стогний А.А., Афанасьев В.Н. Автоматизированные информационные системы. М., "Знание", 1973, № 4.
3. Стогний А.А. Об основных принципах построения автоматизированных информационных систем. - УСМ, 1972, № 2.
4. Стогний А.А., Кондратьев А.И. О нормативной теории игр. - ДАН УССР, 1979, № 9.
5. Кондратьев В.И., Стогний А.А. О понятиях нормативной теории игр. - Кибернетика, 1979, № 4.
6. Журавлев Ю.И. Корректные алгебры над множествами некорректных /эвристических/ алгоритмов I, II, III, Кибернетика 1977, № 4, 1977, № 6, 1978, № 2.
7. *Wald A. Statistical decision functions, Wiley, N.Y., 1950.*
8. Стогний А.А., Кондратьев А.И. Теоретико-игровой подход к решению задач распознавания или классификации. - ДАН УССР, 1980, сер.А., № 4.
9. Кондратьев А.И. Вопросы создания математической теории информационно-распознающих систем. - В кн.: I Всесоюзной конф. по банкам данных (г.Тбилиси, ноябрь 1980 г.): Тез.докл. М. Секция I: изд. ВНИИПОУ, 1980.
10. Стогний А.А., Кондратьев А.И. Информационные системы в управлении (программное обеспечение). - Киев: Знание, 1980.

СПИСК МЕРОПРИЯТИИ

РГ-11 "Системы управления базами данных и информационные системы"

Для выполнения плановых работ РГ-11 провела восемь заседаний, при том участники РГ-11 стимулировали проведение, способствовали организации и принимали участие в мероприятиях, проводимых во время заседаниях РГ-11.

- 1-ое заседание: Берна (Болгария), с 9 по 13 мая 1978 года
Конференция КНВВТ "Применение вычислительных машин и банков данных в научных исследованиях"
- 2-ое заседание: Будапешт (Венгрия), с 29 по 31 января 1979 г.
- 3-ое заседание: Благоевград (Болгария), с 2 по 6 окт. 1979 г.
Конференция КНВВТ и РГ-11 "Вычислительные центры коллективного пользования (ВЦКП)"
- 4-ое заседание: Прага-Либлице (ЧССР), с 24 по 31 мая 1980 г.
Конференция КНВВТ "Применение вычислительной техники в научных исследованиях"
- 5-ое заседание: Тбилиси (СССР), с 11 по 15 ноября 1980 года
Первая всесоюзная конференция "Банки данных"
- 6-ое заседание: Будапешт (Венгрия), с 2 по 8 марта 1981 года
Научная сессия по тематике РГ-11
- 7-ое заседание: Шверин (ГДР), с 13 по 19 декабря 1981 года
Четвертый Международный семинар по банкам данных
- 8-ое заседание: София (Болгария), с 12 по 18 апреля 1982 года
Конференция КНВВТ и РГ-11 "Информационное обеспечение в научных исследованиях"



АДРЕСА УЧАСТНИКОВ

РГ-11 "Системы управления базами данных
и информационные системы"

БАН - БОЛГАРСКАЯ АКАДЕМИЯ НАУК

Барнев, П.	Болгария 1090 София, П.Я.373 Институт математики с ВЦ
Добрев, Д.М.	Болгария 1090 София, П.Я.373 Институт математики с ВЦ
Думков, Пл.	Болгария 1000 София, П.Я.795 БСНИИ "Интерпрограмма"
Иванов, К.	Болгария 1090 София, П.Я.373 Институт математики с ВЦ
Киркова, Р.	Болгария 1090 София, П.Я.373 Институт математики с ВЦ
Эскенази, А.	Болгария 1090 София, П.Я.373 Институт математики с ВЦ

БАН - ВЕНГЕРСКАЯ АКАДЕМИЯ НАУК

Benzur, A.	Hungary 11015 Budapest Csalogany Utca 30-32 MTA SZAMKI
Demetronics, J.	Hungary H-11502 Budapest, POBox 63 MTA SZTAKI

Kiss, O.	Hungary H-1502 Budapest, PO Box 63 MTA SZTAKI
Knuth, E.	Hungary H-1502 Budapest, PO Box 63 MTA SZTAKI
Rado, P.	Hungary H-1502 Budapest, PO Box 63 MTA SZTAKI
Szokolov, M.	Hungary H-1502 Budapest, PO Box 63 MTA SZTAKI

АН ГДР - АКАДЕМИЯ НАУК ГДР

Belke, W.	D D R 2500 Rostock 1 Albert-Einstein-Str. Wilhelm-Pieck Univ./Rechenzentrum
Miller, P.	D D R 1156 Berlin Jacques-Duclos-Str. 50/52 VEB Leitzentrum für Anwendungsforsch.
Öttrich, H.	D D R Mommensenstr. 13 TU Dresden / Sektion 08
Obwald, R.	D D R 1199 Berlin - Adlershof Rudower Chaussee 5 Zentrum für Rechentechnik
Schönian, W.	D D R 1156 Berlin Jacques-Duclos-Str. 50/52 VEB Leitzentrum für Anwendungsforsch.

Schubert, D.	D D R 8027 Dresden Mommssenstr.13 TU Dresden / Sektion 08
Werner, W.	D D R 1199 Berlin - Adlershof Rudower Chaussee 5 Zentrallinstitut für Kybernetik

АНР - АКАДЕМИЯ НАУК РЕСПУБЛИКИ КУБА

Dipotet, P.	Cuba Habana Academie de Ciencias, IMACC
-------------	---

ПАН - ПОЛЬСКАЯ АКАДЕМИЯ НАУК

Bankowski, J.	Poland 00-950 Warszawa Al.Niepodleglosci 188 I I N T E
Dabrowsky, M.	Poland 00-950 Warszawa Al.Niepodleglosci 188 I I N T E
Gorny, A.	Poland 41-200 Sosnowiec Ul.Zeromskiego 3 Inst.Informacji Naukowo-Technicznej
Rzeczkowski, W.	Poland 00-901 Warszawa, PKiN Instytut Podstaw Informatyki
Rudnicki, P.	Poland 00-950 Warszawa Al.Niepodleglosci 188 I I N T E

Subieta, K.	Poland 00-901 Warszawa, PKiN Inst. Podstaw Informatyki
Wakulicz, A.	Poland 41-200 Sosnowiec Ul. Zeromskiego 3 Inst. Informacji Naukowo-Technicznej

АН СССР - АКАДЕМИЯ НАУК СССР

Кондратьев, А.И.	СССР 252207 Киев-207 Пр. 40-летия окт., 142/144 Институт кибернетики АН УССР
Савинков, В.М.	СССР 111024 Москва Авиамоторная ул., 26/5 ВНИИПОУ ГКНТ
Столгний, А.А.	СССР 252207 Киев-207 Пр. 40-летия окт., 142/144 Институт кибернетики АН УССР
Столяров, Г.К.	СССР 220604 Минск-72 ул. Сурганова 11 Институт математики АН БССР

ЧСАН - ЧЕХОСЛОВАЦКАЯ АКАДЕМИЯ НАУК

Havel, I.	Czechoslovakia 11567 Praha-1 Zitna 25 Matematicky Ustav CSAV
Liebl, P.	Czechoslovakia 11567 Praha-1 Zitna 25 Matematicky Ustav CSAV

Riha, A.

Czechoslovakia

11800 Praha-1

Malostranske Nam. 25

V C U K

ЦНИИ - ЦЕНТР НАУЧНЫХ ИССЛЕДОВАНИЙ ВЬЕТНАМА

Ho Thuan

Viet-Nam

Ha-Noi

VIEN KHOA Viet-Nam

A TANULMÁNSOROZATBAN 1981–BEN MEGJELENTEK:

- 116/1981 Sieglér András: Egy 6 szabadságfokú antropomorf manipulátor kinematikája számítógépes vezérlése
- 117/1981 Knuth Előd – Radó Péter: Principles of Computer Aided System Description
- 118/1981 Demetrovics János – Gyepesi György: Általános függőségek és lekérdezéssel kapcsolatos algoritmusok relációs adatmodellekben
- 119/1981 Sztanó Tamás: REAL–TIME programrendszerek eseményvezérelt szervezése
- 120/1981 Szentgyörgyi Zsuzsa: A számítástechnika műszaki fejlődése és társadalmi hatásai
- 121/1981 Vicsek Tamásné (Strehó Mária) : Vizsgálatok a kezdeti érték problémák numerikus megoldásával kapcsolatban
- 122/1981 Andó Györgyi – Lipcsey Zsolt: Sztochasztikus Ljapunov módszerek és alkalmazásaik
- 123/1981 Márkus Zsuzsanna: Intelligens interaktív rendszerek elvi problémái
- 124/1981 Márkus Zsuzsanna: Logikai alapú programozási módszerek és alkalmazásaik számítógéppel segített építészeti tervezési feladatok megoldásához
- 125/1981 Fabók Julianna: Software implementációs nyelvek
- 126/1981 Várszegi Sándor: Multimikroszámítógépes-rendszerek
- 127/1981 Lipcsey Zsolt: N-személyes minőségi differenciáljátékok késleltetéssel és késleltetés nélkül
- 128/1981 Böszörményi László: Multa-task rendszerek fejlesztése magasszintű nyelven
- 129/1981 Tóth János: A formális reakciókinetika globális determinisztikus és sztochasztikus modelljéről és néhány alkalmazásáról



